ACL-05

Building and Using Parallel Texts:

Data-Driven Machine Translation and Beyond

Proceedings of the Workshop

29-30 June 2005
University of Michigan
Ann Arbor, Michigan, USA

Production and Manufacturing by
Omnipress Inc.
Post Office Box 7214
Madison, WI 53707-7214

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
75 Paterson Street, Suite 9
New Brunswick, NJ 08901
USA
Tel: +1-732-342-9100
Fax: +1-732-342-9339
acl@aclweb.org

Introduction

The ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond, took place on Tuesday, June 29 and Wednesday, June 30 in Ann Arbor Michigan, immediately following the 43rd Annual Meeting of the Association for Computational Linguistics.

This workshop represented a merger of two workshops that were originally proposed as independent events. Joel Martin, Rada Mihalcea, and Ted Pedersen had proposed a workshop on *Building and Using Parallel Texts for Languages with Scarce Resources*, which was intended as a follow-up event to the NAACL 2003 Workshop on Parallel Text that had been organized by Mihalcea and Pedersen. At the same time, Philipp Koehn and Christof Monz had proposed a workshop on *Exploiting Parallel Texts for Statistical Machine Translation*, featuring a shared task on Phrase Based Machine Translation.

Given the close relationship between the two proposed topics, the idea of a merger was quickly embraced by all concerned. It was agreed that the workshop would have two tracks, one regarding Parallel Texts for Languages with Scarce Resources (Track 1), and the other focused on Statistical Machine Translation (Track 2).

Prior to the workshop, in addition to soliciting relevant papers for review and possible presentation, the organizers of both tracks conducted shared tasks that brought together systems for an evaluation on previously unseen data. Track 1 featured a Word Alignment shared task, where the object was to align parallel text in one or more of the following langauge pairs: Inuktitut–English, Romanian–English, and Hindi–English. Track 2 carried out a shared task on Phrase Based Statistical Machine Translation, where eleven participating teams competed to build machine translation systems for French–English, Spanish–English, German–English, and Finnish–English.

The results of the shared tasks were announced at the workshop, and these proceedings also include an overview paper for each shared task that summarizes the results, as well as provides information about the data used and any procedures that were followed in conducting or scoring the task. In addition, there are short papers from each participating team for each shared task that describe their underlying system in some detail.

Tuesday June 29 was dedicated to Track 1. It featured an invited talk by Mike Maxwell of the Linguistic Data Consortium, eight long paper presentations relevant to the topic of building and using parallel texts for languages with scarce resources, six short paper presentations describing systems that participated in the Word Alignment shared task (four additional short papers are included in the proceedings), a shared task overview, and a panel discussion about lessons learned from the shared task.

Track 2 was featured on Wednesday June 30. It included an invited talk by Franz Josef Och of Google, six long paper presentations, a shared task overview, and nine shared task system descriptions.

We would like to thank the members of the Program Committee for their timely reviews.

Philipp Koehn, Joel Martin, Rada Mihalcea, Christof Monz, and Ted Pedersen
Co–Organizers

Organizers:

Philipp Koehn (University of Edinburgh)
Joel Martin (National Research Council of Canada)
Rada Mihalcea (University of North Texas)
Christof Monz (University of Maryland)
Ted Pedersen (University of Minnesota, Duluth)

Invited Speakers:

Mike Maxwell (Linguistic Data Consortium, University of Pennsylvania)
Franz Josef Och (Google)

Program Committee:

Lars Ahrenberg (Linköping University)
Bill Byrne (University of Cambridge, Johns Hopkins University)
Chris Callison-Burch (University of Edinburgh)
Nicoletta Calzolari (Istituto di Linguistica Computazionale del CNR, Pisa)
Francisco Casacuberta (Universitat Politècnica de València)
David Chiang (University of Maryland)
Mona Diab (Columbia University)
George Foster (National Research Council of Canada)
Alexander Fraser (ISI/University of Southern California)
Pascale Fung (Hong Kong University of Science and Technology)
Rob Gaizauskas (University of Sheffield)
Ulrich Germann (University of Toronto)
Dan Gildea (University of Rochester)
Jan Hajic (Charles University)
Andrew Hardie (University of Lancaster)
Rebecca Hwa (University of Pittsburgh)
Nancy Ide (Vassar College)
Kevin Knight (ISI/University of Southern California)
Greg Kondrak (University of Alberta)
Roland Kuhn (National Research Council of Canada)
Shankar Kumar (Johns Hopkins University)
Philippe Langlais (University of Montreal)
Alon Lavie (Carnegie Mellon University)
Lori Levin (Carnegie Mellon University)
Daniel Marcu (ISI/University of Southern California)
Tony McEnery (University of Lancaster)
Bridget McInnes (University of Minnesota, Twin Cities)
Magnus Merkel (Linköping University)
Bob Moore (Microsoft Research)

Herman Ney (RWTH Aachen)
Maria das Graças Volpe Nunes (University of São Paulo)
Franz Josef Och (Google)
Kemal Oflazer (Sabancı University)
Miles Osborne (University of Edinburgh)
Andrei Popescu-Belis (University of Geneva)
Katharina Probst (Carnegie Mellon University)
Amruta Purandare (University of Pittsburgh)
Florence Reeder (MITRE)
Philip Resnik (University of Maryland)
Antonio Ribeiro (European Commission, Joint Research Centre)
Michel Simard (Xerox Research Centre Europe)
Kevin Scannell (St. Louis University)
Libin Shen (University of Pennsylvania)
Eiichiro Sumita (ATR Spoken Language Communication Research Laboratories)
Joerg Tiedemann (University of Groningen)
Christoph Tillmann (IBM)
Hajime Tsukada (NTT Communication Science Laboratories)
Dan Tufiş (Research Institute for AI of the Romanian Academy)
Jean Véronis (Université de Provence)
Michelle Vanni (Army Research Lab)
Stephan Vogel (Carnegie Mellon University)
Clare Voss (Army Research Lab)
Taro Watanabe (ATR Spoken Language Translation Research Laboratories)
Dekai Wu (Hong Kong University of Science and Technology)

Additional Reviewers:

Colin Cherry (University of Alberta)
Behrang Mohit (University of Pittsburgh)

Table of Contents

Conference Program

Wednesday, June 29, 2005

8:45–9:00 Welcome

Invited Talk

9:00–10:00 Mike Maxwell *So many languages, so few resources: How to bridge the gap?*

Session 1: Long Papers

10:00–10:20 *Association-Based Bilingual Word Alignment*
Robert C. Moore

10:20–11:00 Break

Session 2: Long Papers (continued)

11:00–11:20 *Cross Language Text Categorization by Acquiring Multilingual Domain Models from Comparable Corpora*
Alfio Gliozzo and Carlo Strapparava

11:20–11:40 *Parsing Word-Aligned Parallel Corpora in a Grammar Induction Context*
Jonas Kuhn

11:40–12:00 *Bilingual Word Spectral Clustering for Statistical Machine Translation*
Bing Zhao, Eric P. Xing and Alex Waibel

12:00–12:20 *Revealing Phonological Similarities between Related Languages from Automatically Generated Parallel Corpora*
Karin Müller

12:20–12:40 *Augmenting a Small Parallel Text with Morpho-Syntactic Language*
Maja Popović, David Vilar, Hermann Ney, Slobodan Jovičić and Zoran Šarić

12:40–2:00 Lunch

Wednesday, June 29, 2005 (continued)

Session 3: Long Papers (continued)

2:00–2:20 *Induction of Fine-Grained Part-of-Speech Taggers via Classifier Combination and Crosslingual Projection*
Elliott Drábek and David Yarowsky

2:20–2:40 *A Hybrid Approach to Align Sentences and Words in English-Hindi Parallel Corpora*
Niraj Aswani and Robert Gaizauskas

Shared Task I Overview

2:40–3:00 *Word Alignment for Languages with Scarce Resources*
Joel Martin, Rada Mihalcea and Ted Pedersen

Session 4: Shared Task I Papers

3:00–3:15 *NUKTI: English-Inuktitut Word Alignment System Description*
Philippe Langlais, Fabrizio Gotti and Guihong Cao

3:15–3:30 *Models for Inuktitut-English Word Alignment*
Charles Schafer and Elliott Drábek

3:30–4:00 Break

Session 5: Shared Task I Papers (continued)

4:00–4:15 *Improved HMM Alignment Models for Languages with Scarce Resources*
Adam Lopez and Philip Resnik

4:15–4:30 *Symmetric Probabilistic Alignment*
Ralf D. Brown, Jae Dong Kim, Peter J. Jansen and Jaime G. Carbonell

4:30–4:45 *ISI's Participation in the Romanian-English Alignment Task*
Alexander Fraser and Daniel Marcu

4:45–5:00 *Experiments Using MAR for Aligning Corpora*
Juan Miguel Vilar

Wednesday, June 29, 2005 (continued)

Shared Task I Papers without Presentations

Comparison, Selection and Use of Sentence Alignment Algorithms for New Language Pairs
Anil Kumar Singh and Samar Husain

Combined Word Alignments
Dan Tufis, Radu Ion, Alexandru Ceausu and Dan Stefanescu

LIHLA: Shared Task System Description
Helena M. Caseli, Maria G. V. Nunes and Mikel L. Forcada

Aligning words in English-Hindi Parallel Corpora
Niraj Aswani and Robert Gaizauskas

Shared Task I Panel Discussion

5:00–6:00 TBA *Lessons Learned, and Future Directions*

Thrusday, June 30, 2005

Shared Task II Overview

9:15–9:30 *Shared Task: Statistical Machine Translation between European Languages*
Philipp Koehn and Christof Monz

Session 6: Shared Task II Papers

9:30–9:45 *Improved Language Modeling for Statistical Machine Translation*
Katrin Kirchhoff and Mei Yang

9:45–10:00 *PORTAGE: A Phrase-Based Machine Translation System*
Fatiha Sadat, Howard Johnson, Akakpo Agbago, George Foster, Roland Kuhn, Joel Martin and Aaron Tikuisis

10:00–10:15 *Statistical Machine Translation of Euparl Data by using Bilingual N-grams*
Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert and José B. Mariño

Thrusday, June 30, 2005 (continued)

10:15–11:00 Break

Session 7: Shared Task II Papers (continued)

11:00–11:15 *RALI: SMT Shared Task System Description*
Philippe Langlais, Guihong Cao and Fabrizio Gotti

11:15–11:30 *A Generalized Alignment-Free Phrase Extraction*
Bing Zhao and Stephan Vogel

11:30–11:45 *Combining Linguistic Data Views for Phrase-based SMT*
Jesús Giménez and Lluís Màrquez

11:45–12:00 *Improving Phrase-Based Statistical Translation by Modifying Phrase Extraction and Including Several Features*
Marta Ruiz Costa-jussà and José A. R. Fonollosa

12:00–12:15 *First Steps towards Multi-Engine Machine Translation*
Andreas Eisele

12:15–12:30 *Competetive Grouping in Integrated Phrase Segmentation and Alignment Model*
Ying Zhang and Stephan Vogel

Shared Task II Paper without Presentation

Deploying Part-of-Speech Patterns to Enhance Statistical Phrase-Based Machine Translation Resources
Christina Lioma and Iadh Ounis

12:30–2:00 Lunch

Thrusday, June 30, 2005 (continued)

Invited Talk

2:00–3:00 Franz Och *Statistical Machine Translation: The Fabulous Present and Future*

Session 7: Long Papers

3:10–3:30 *Novel Reordering Approaches in Phrase-Based Statistical Machine Translation*
Stephan Kanthak, David Vilar, Evgeny Matusov, Richard Zens and Hermann Ney

3:30–4:00 Break

Session 8: Long Papers (continued)

4:00–4:20 *Gaming Fluency: Evaluating the Bounds and Expectations of Segment-based Translation Memory*
John Henderson and William Morgan

4:20–4:40 *Hybrid Example-Based SMT: the Best of Both Worlds?*
Declan Groves and Andy Way

4:40–5:00 *Word Graphs for Statistical Machine Translation*
Richard Zens and Hermann Ney

5:00–5:20 *A Recursive Statistical Translation Model*
Juan Miguel Vilar and Enrique Vidal

5:20–5:40 *Training and Evaluating Error Minimization Decision Rules for Statistical Machine Translation*
Ashish Venugopal, Andreas Zollmann and Alex Waibel

Association-Based Bilingual Word Alignment

Robert C. Moore
Microsoft Research
One Microsoft Way
Redmond, WA 98052
bobmoore@microsoft.com

Abstract

Bilingual word alignment forms the foundation of current work on statistical machine translation. Standard word-alignment methods involve the use of probabilistic generative models that are complex to implement and slow to train. In this paper we show that it is possible to approach the alignment accuracy of the standard models using algorithms that are much faster, and in some ways simpler, based on basic word-association statistics.

1 Motivation

Bilingual word alignment is the first step of most current approaches to statistical machine translation. Although the best performing systems are "phrase-based" (see, for instance, Och and Ney (2004) or Koehn et al. (2003)), possible phrase translations must first be extracted from word-aligned bilingual text segments. The standard approach to word alignment makes use of five translation models defined by Brown et al. (1993), sometimes augmented by an HMM-based model or Och and Ney's "Model 6" (Och and Ney, 2003). The best of these models can produce high accuracy alignments, at least when trained on a large parallel corpus of fairly direct translations in closely related languages.

There are a number of ways in which these standard models are less than ideal, however. The higher-accuracy models are mathematically complex, and also difficult to train, as they do not factor in a way that permits a dynamic programming solution. It can thus take many hours of processing time on current standard computers to train the models and produce an alignment of a large parallel corpus.

In this paper, we take a different approach to word alignment, based on the use of bilingual word-association statistics rather than the generative probabilistic framework that the IBM and HMM models use. In the end we obtain alignment algorithms that are much faster, and in some ways simpler, whose accuracy comes surprisingly close to the established probabilistic generative approach.

2 Data and Methodology for these Experiments

The experiments reported here were carried out using data from the workshop on building and using parallel texts held at HLT-NAACL 2003 (Mihalcea and Pedersen, 2003). For the majority of our experiments, we used a subset of the Canadian Hansards bilingual corpus supplied for the workshop, comprising 500,000 English-French sentences pairs, including 37 sentence pairs designated as "trial" data, and 447 sentence pairs designated as test data. The trial and test data have been manually aligned at the word level, noting particular pairs of words either as "sure" or "possible" alignments. As an additional test, we evaluated our best alignment method using the workshop corpus of approximately 49,000 English-Romanian sentences pairs from diverse sources, including 248 manually aligned sentence pairs designated as test data.[1]

[1]For the English-French corpus, automatic sentence alignment of the training data was provided by Ulrich Germann,

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 1–8,
Ann Arbor, June 2005.

We needed annotated development data to optimize certain parameters of our algorithms, and we were concerned that the small number of sentence pairs designated as trial data would not be enough for this purpose. We therefore randomly split each of the English-French and English-Romanian test data sets into two virtually equal subsets, by randomly ordering the test data pairs, and assigning alternate pairs from the random order to the two subsets. We used one of these subsets as a development set for parameter optimization, and held out the other for a final test set.

We report the performance of various alignment algorithms in terms of precision, recall, and alignment error rate (AER) as defined by Och and Ney (2003):

$$\text{recall} = \frac{|A \cap S|}{|S|}$$

$$\text{precision} = \frac{|A \cap P|}{|A|}$$

$$\text{AER} = 1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$$

In these definitions, S denotes the set of alignments annotated as sure, P denotes the set of alignments annotated possible or sure, and A denotes the set of alignments produced by the method under test. Following standard practice in the field, we take AER, which is derived from F-measure, as the primary evaluation metric that we are attempting to optimize.

Our initial experiments involve algorithms that do not consider the positions of words in the sentences. Thus, they are incapable of distinguishing among multiple instances of the same word type in a sentence. We will say that these methods produce word type alignments. We compare these algorithms on the basis of the best possible alignment of word tokens given an alignment of word types. We go on to consider various ways of choosing a word token alignment for a given word type alignment, and all our final evaluations are conducted on the basis of the alignment of individual word tokens.

and the hand alignments of the words in the trial and test data were created by Franz Och and Hermann Ney (Och and Ney, 2003). The manual word alignments for the English-Romanian test data were created by Rada Mihalcea and Ted Pedersen.

3 The Log-Likelihood-Ratio Association Measure

We base all our association-based word-alignment methods on the log-likelihood-ratio (LLR) statistic introduced to the NLP community by Dunning (1993). We chose this statistic because it has previously been found to be effective for automatically constructing translation lexicons (e.g., Melamed, 2000). We compute LLR scores using the following formula presented by Moore (2004):

$$LLR(f, e) = \sum_{f? \in \{f, \neg f\}} \sum_{e? \in \{e, \neg e\}} C(f?, e?) \log \frac{p(f?|e?)}{p(f?)}$$

In this formula f and e mean that the words whose degree of association is being measured occur in the respective target and source sentences of an aligned sentence pair, $\neg f$ and $\neg e$ mean that the corresponding words do not occur in the respective sentences, $f?$ and $e?$ are variables ranging over these values, and $C(f?, e?)$ is the observed joint count for the values of $f?$ and $e?$. The probabilities in the formula refer to maximum likelihood estimates.

Since the LLR score for a pair of words is high if the words have either a strong positive association or a strong negative association, we discard any negatively associated word pairs by requiring that $p(f, e) > p(f) \cdot p(e)$. Initially, we computed the LLR scores for all positively associated English/French word pairs in our 500K sentence pair corpus. To reduce the memory requirements of our algorithms we discarded any word pairs whose LLR score was less than 1.0. This left us with 12,797,697 word pairs out of a total of 21,451,083 pairs that had at least one co-occurrence.

4 One-to-One, Word Type Alignment Methods

4.1 Method 1

The first set of association-based word-aligment methods we consider permit only one-to-one alignments and do not take word position into account. The simplest method we consider uses the LLR scores to link words according to Melamed's (2000) "competitive linking algorithm" for aligning words in a pair of sentences. Since competitive linking has

no way to distinguish one instance of a particular word type from another, we operate with counts of linked and unlinked instances of word types, without trying to designate the particular instances the counts refer to. This version of competitive linking can be described as follows:

- Find the pair consisting of an English word type and a French word type that have the highest association score of any pair of words types that both have remaining unlinked instances.
- Increase by 1 the count of linked occurrences of this pair of word types, and decrease by 1 the count of unlinked instances of each of these word types.
- Repeat until no more words can be linked.

We will refer to this version of the competitive linking algorithm using LLR scores as Method 1. This is the method that Melamed uses to generate an initial alignment that he refines by re-estimation in his "Method A" (Melamed, 2000).

Method 1 can terminate either because one or both sentences of the pair have no more unlinked words, or because no association scores exist for the remaining unlinked words. We can use this fact to trade off recall for precision by discarding association scores below a given threshold. Table 1 shows the precision/recall trade-off for Method 1 on our development set. Since Method 1 produces only word type alignments, these recall and precision scores are computed with respect to an oracle that makes the best possible choice among multiple occurrences of the same word type.[2] The best (oracular) AER is 0.216, with recall of 0.840 and precision of 0.747, occurring at an LLR threshold of 11.7.

4.2 Method 2

A disadvantage of Method 1 is that it makes alignment decisions for each sentence pair independently of the decisions for the same words in other sentence pairs. It turns out that we can improve alignment accuracy by biasing the alignment method towards linking words in a given sentence that are also linked in many other sentences. A simple way to do this is to perform a second alignment based on the conditional probability of a pair of words being linked according to Method 1, given that they both occur in a given sentence pair. We estimate this link probability LP as

$$LP(f,e) = \frac{links_1(f,e)}{cooc(f,e)}$$

where $links_1(f,e)$ is the number of times f and e are linked according to Method 1, and $cooc(f,e)$ is the number of times f and e co-occur in aligned sentences.[3]

We now define alignment Method 2 as follows:

- Count the number of links in the training corpus for each pair of words linked in any sentence pair by Method 1.
- Count the number of co-occurrences in the training corpus for each pair of words linked in any sentence pair by Method 1.
- Compute LP scores for each pair of words linked in any sentence pair by Method 1.
- Align sentence pairs by competitive linking using LP scores.

Recall	Precision	Threshold
0.111	0.991	168368
0.239	0.923	71074
0.304	0.902	53286
0.400	0.838	26001
0.501	0.822	11306
0.600	0.788	4224
0.700	0.778	1141
0.800	0.765	124
0.848	0.732	1

Table 1: Recall/Precision Trade-Off for Method 1.

[2] The oracle goes through the word type pairs in the same order as the competitive linking algorithm, linking particular instances of the word types. It prefers a pair that has a sure alignment in the annotated test data to a pair that has a possible alignment; and prefers a pair with a possible alignment to one with no alignment.

[3] Melamed (1998) points out there are at least three ways to count the number of co-ccurrences of f and e in a given sentence pair if one or both of f and e have more than one occurrence. Based on preliminary explorations, we chose to count the co-occurrences of f and e as the maximum of the number of occurrences of f and the number of occurrences of e, if both f and e occur; otherwise $cooc(f,e) = 0$.

Recall	Precision	Threshold
0.100	0.887	0.989
0.230	0.941	0.982
0.301	0.952	0.967
0.400	0.964	0.938
0.501	0.967	0.875
0.600	0.967	0.811
0.705	0.948	0.649
0.816	0.921	0.441
0.880	0.775	0.000

Table 2: Recall/Precision Trade-Off for Method 2.

Recall	Precision	Threshold
0.178	1.000	0.982
0.200	0.998	0.977
0.300	0.999	0.958
0.405	0.998	0.923
0.502	0.994	0.871
0.602	0.987	0.758
0.737	0.947	0.647
0.804	0.938	0.441
0.883	0.776	0.000

Table 3: Recall/Precision Trade-Off for Method 3.

Table 2 shows the precision/recall trade-off for Method 2 on our development set. Again, an oracle is used to choose among multiple occurrences of the same word type. The best (oracular) AER is 0.126, with recall of 0.830 and precision of 0.913, occurring at an LP threshold of 0.215.

4.3 Method 3

It is apparent that Method 2 performs much better than Method 1 at any but the lowest recall levels. However, it fails to display a monotonic relationship between recall and precision as the score cut-off threshold is tightened or loosened. This seems to be due to the fact that the LP measure, unlike LLR, does not discount estimates made on the basis of little data. Thus a pair of words that has one co-occurrence in the corpus, which is linked by Method 1, gets the same LP score of 1.0 as a pair of words that have 100 co-occurrences in the corpus and are linked by Method 1 every time they co-occur.

A simple method of compensating for this over-confidence in rare events is to apply absolute discounting. We will define the discounted link probability LP_d similarly to LP, except that a fixed discount d is subtracted from each link count:

$$LP_d(f,e) = \frac{links_1(f,e) - d}{cooc(f,e)}$$

Method 3 is then identical to Method 2, except that LP_d is used in place of LP. We determined the optimal value of d for our development set to be approximately 0.9, using the optimal, oracular AER as our objective function.

Table 3 shows the precision/recall trade-off for Method 3 on our development set, with $d = 0.9$ and use of an oracle to choose among multiple occurrences of the same word type. The best (oracular) AER is 0.119, with recall of 0.827 and precision of 0.929, occurring at an LP_d threshold of 0.184. This is an improvement of 0.7% absolute in AER, but perhaps as importantly, the monotonic trade-off between precision and recall is essentially restored. We can see in Table 3 that we can achieve recall of 60% on this development set with precision of 98.7%, and we can obtain even higher precision by sacrificing recall slightly more. With Method 2, 96.7% was the highest precision that could be obtained at any recall level measured.

5 Allowing Many-to-One Alignments

It appears from the results for Methods 2 and 3 on the development set that reasonable alignment accuracy may be achievable using association-based techniques (pending a way of selecting the best word token alignments for a given word type alignment). However, we can never learn any many-to-one alignments with methods based on competitive linking, as either we or Melamed have used it so far.

To address this issue, we introduce the notion of bilingual word clusters and show how iterated applications of variations of Method 3 can learn many-to-one mappings by building up clusters incrementally. Consider the abstract data structure to which competitive linking is applied as a tuple of bags (multi-sets). In Methods 1–3, for each sentence pair, competitive linking is applied to a tuple of a bag of French words and a bag of English words. Suppose we apply Method 3 with a high LP_d cut-off threshold so that we can be confident that almost all

the links we produce are correct, but many French and English words remain unlinked. We can regard this as producing for each sentence pair a tuple of three bags: bags of the remaining unlinked English and French words, plus a third bag of word clusters consisting of the linked English and French words. To produce more complex alignments, we can then carry out an iteration of a generalized version of Method 3, in which competitive linking connects remaining unlinked English and French words to each other or to previously derived bilingual clusters.[4]

As just described, the approach does not work very well, because it tends to build clusters too often when it should produce one-to-one alignments. The problem seems to be that translation tends to be nearly one-to-one, especially with closely related languages, and this bias is not reflected in the method so far. To remedy this, we introduce two biases in favor of one-to-one alignments. First, we discount the LLR scores between words and clusters, so the competitive linking pass using these scores must find a substantially stronger association for a given word to a cluster than to any other unlinked word before it will link the word to the cluster. Second, we apply the same high LP_d cut-off on word-to-cluster links that we used in the first iteration of Method 3 to generate word-to-word links. This leaves many unlinked words, so we apply one more iteration of yet another modified version of Method 3 in which competitive linking is allowed to link the remaining unlinked words to other unlinked words, but not to clusters. We refer to this sequence of three iterations of variations of Method 3 as Method 4.

To evaluate alignments involving clusters according Och and Ney's method, we translate clusters back into all possible word-to-word alignments consistent with the cluster. We found the optimal value on the development set for the LLR discount for clusters to be about 2000, and the optimal value for the LP_d cut-off for the first two iterations of Method 3 to be about 0.7. With these parameter values, the best (oracular) AER for Method 4 is 0.110, with recall of 0.845 and precision of 0.929, occurring at a final LP_d threshold of 0.188. This is an improvement of 0.9% absolute in AER over Method 3, resulting from an improvement of 1.7% absolute in recall, with virtually no change in precision.

[4] In principle, the process can be further iterated to build up clusters of arbitrary size, but at this stage we have not yet found an effective way of deciding when a cluster should be expanded beyond two-to-one or one-to-two.

6 Token Alignment Selection Methods

Finally, we turn to the problem of selecting the best word token alignment for a given word type alignment, and more generally to the incorporation of positional information into association-based word-alignment. We consider three token alignment selection methods, each of which can be combined with any of the word type alignment methods we have previously described. We will therefore refer to these methods by letter rather than number, with a complete word token alignment method being designated by a number/letter combination.

6.1 Method A

The simplest method for choosing a word token alignment for a given word type alignment is to make a random choice (without replacement) for each word type in the alignment from among the tokens of that type. We refer to this as Method A.

6.2 Method B

In Method B, we find the word token alignment consistent with a given word type alignment that is the most nearly mononotonic. We decide this by defining the degree of nonmonotonicity of an alignment, and minimizing that. If more than one word token alignment has the lowest degree of nonmonotonicity, we pick one of them arbitrarily.

To compute the nonmonotonicity of a word token alignment, we arbitrarily designate one of the languages as the source and the other as the target. We sort the word pairs in the alignment, primarily by source word position, and secondarily by target word position. We then iterate through the sorted alignment, looking only at the target word positions. The nonmonotonicity of the alignment is defined as the sum of the absolute values of the backward jumps in this sequence of target word positions.

For example, suppose we have the sorted alignment ((1,1)(2,4)(2,5)(3,2)). The sequence of target word positions in this sorted alignment is (1,4,5,2). This has only one backwards jump, which is of size 3, so that is the nonmonotonicity value for this alignment. For a complete or partial alignment, the

nonmonotonicity is clearly easy to compute, and nonmonotonicity can never be decreased by adding links to a partial alignment. The least nonmonotonic alignment is found by an incremental best-first search over partial alignments kept in a priority queue sorted by nonmonotonicity.

6.3 Method C

Method C is similiar to Method B, but it also uses nonmonotonicity in deciding which word types to align. In Method C, we modify the last pass of competitive linking of the word type alignment method to stop at a relatively high score threshold, and we compute all minimally nonmonotonic word token alignments for the resulting word type alignment.

We then continue the final competitive linking pass applied to word tokens rather than types, but we select only word token links that can be added to one of the remaining word token alignments without increasing its nonmonotonicity. Specifically, for each remaining word type pair (in order of decreasing score) we make repeated passes through all of the word token alignments under consideration, adding one link between previously unlinked instances of the two word types to each alignment where it is possible to do so without increasing nonmonotonicity, until there are no longer unlinked instances of both word types or no more links between the two word types can be added to any alignment without increasing its nonmonotonicity. At the end of each pass, if some, but not all of the alignments have had a link added, we discard the alignments that have not had a link added; if no alignments have had a link added, we go on to the next word type pair. This final competitive linking pass continues until another, lower score threshold is reached.

6.4 Comparison of Token Alignment Selection Methods

Of these three methods, only C has additional free parameters, which we jointly optimized on the development set for each of the word type alignment methods. All other parameters were left at their optimal values for the oracular choice of word token alignment.

Table 4 shows the optimal AER on the development set, for each combination of word type alignment method and token alignment selection method that we have described. For comparison, the oracle for each of the pure word type alignment methods is added to the table as a token alignment selection method. As we see from the table, Method 4 is the best word type alignment method for every token alignment selection method, and Method C is the best actual token alignment selection method for every word type alignment method. Method C even beats the token alignment selection oracle for every word alignment type method except Method 1. This is possible because Method C incorporates nonmonotonicity information into the selection of linked word types, whereas the oracle is applied after all word type alignments have been chosen.

	Oracle	A	B	C
1	0.216	0.307	0.255	0.243
2	0.126	0.210	0.147	0.109
3	0.119	0.208	0.138	0.103
4	0.110	0.196	0.130	0.098

Table 4: Development Set AER for all Methods.

The best combined overall method is 4C. For this combination, the optimal value on the development set for the first score threshold of Method C was about 0.65 and the optimal value of the second score threshold of Method C was about 0.075.

7 Evaluation

We computed the recall, precision, and AER on the held-out subset of the English-French data both for our Method 4C (using parameter values optimized on the development subset) and for IBM Model 4, computed using Och's Giza++ software package (Och and Ney, 2003) trained on the same data as Method 4C. We used the default configuration file included with the version of Giza++ that we used, which resulted in five iterations of Model 1, followed by five iterations of the HMM model, followed by five iterations of Model 4. We trained and evaluated the models in both directions, English-to-French and French-to-English, as well as the union, intersection, and what Och and Ney (2003) call the "refined" combination of the two alignments. The results are shown in Table 5. We applied the same evaluation methodology to the English-Romanian data, with the results shown in Table 6.

Alignment	Recall	Precision	AER
Method 4C	0.879	0.929	0.094
E → F	0.870	0.890	0.118
F → E	0.876	0.907	0.106
Union	0.929	0.845	0.124
Intersection	0.817	0.981	0.097
Refined	0.908	0.929	0.079

Table 5: English-French Results.

Alignment	Recall	Precision	AER
Method 4C	0.580	0.881	0.301
E → R	0.545	0.759	0.365
R → E	0.549	0.741	0.370
Union	0.570	0.423	0.515
Intersection	0.180	0.901	0.820
Refined	0.584	0.759	0.328

Table 6: English-Romanian Results.

Comparison of the AER for Method 4C and IBM Model 4 shows that, in these experiments, only the refined combination of both directions of the Model 4 alignments outperforms our method, and only on the English-French data (and by a relatively small amount: 16% relative reduction in error rate). Our existing Perl implementation of Method 4C takes about 3.5 hours for the 500K sentence pair data set on a standard desk top computer. It took over 8 hours to train each direction of Model 4 using Giza++ (which is written in C++). We believe that if our method was ported to C++, our speed advantage over Giza++ would be substantially greater. Previous experience porting algorithms of the same general type as Method 4C from Perl to C++ has given us speed ups of a factor of 10 or more.

Note that we were unable to optimize the many options and free parameters of Giza++ on the development data, as we did with the parameters of Method 4C, which perhaps inhibits us from drawing stronger conclusions from these experiments. However, it was simply impractical to do so, due the time required to re-train the Giza++ models with new settings. With Method 4C, on the other hand, most of the time is spent either in computing initial corpus statistics that are independent of the parameters settings, or in performing the final corpus alignment once the parameters settings have been optimized. Of the five parameters Method 4C requires, changes to three of them took less than one hour of retraining (on the English-French data – much less on the English-Romanian data), and settings of the last two need to be tested only on the small amount of annotated development data, which took only a few seconds. This made it possible to optimize the parameters of Method 4C in a small fraction of the time that would have been required for Giza++.

8 Related Work

The literature on measures of bilingual word association is too large to review thoroughly, but mostly it concerns extracting bilingual lexicons rather than word alignment. We discuss three previous research efforts that seem particularly relevant here.

Gale and Church (1991) made what may be the first application of word association to word alignment. Their method seems somewhat like our Method 1B. They use a word association score directly, although they use the ϕ^2 statistic instead of LLR, and they consider forward jumps as well as backward jumps in a probability model in place of our nonmonotonicity measure. They report 61% recall at 95% precision on Canadian Hansards data.

Obviously, we are building directly on the work of Melamed (2000), sharing his use of the LLR statistic and adopting his competitive linking algorithm. We diverge in other details, however. Moreover, Melamed makes no provision for other than one-to-one alignments, and he does not deal with the problem of turning a word type alignment into a word token alignment. As Table 4 shows, this is crucial to obtaining high accuracy alignments.

Finally, our work is similar to that of Cherry and Lin (2003) in our use of the conditional probability of a link given the co-occurrence of the linked words. Cherry and Lin generalize this idea to incorporate additional features of the aligned sentence pair into the conditioning information. The chief difference between their work and ours, however, is their dependence on having parses for the sentences in one of the languages being aligned. They use this to enforce a phrasal coherence constraint, which basically says that word alignments cannot cross constituent boundaries. They report excellent alignment

accuracy using this approach, and one way of comparing our results to theirs is to say that we show it is also possible to get good results (at least for English and French) by using nonmonotonicity information in place of constituency information.

9 Conclusions

The conventional wisdom in the statistical MT community has been that "heuristic" alignment methods based on word association statistics could not be competitive with methods that have a "well-founded mathematical theory that underlies their parameter estimation" (Och and Ney, 2003, p. 37). Our results seem to suggest that this is not the case. While we would not claim to have demonstated that association-based methods are superior to the established approach, they certainly now appear to be worth investigating further.

Moreover, our alignment method is faster than standard models to train; potentially much faster if it were re-implemented in a language like C++. Efficiency issues, especially in training, are often dismissed as unimportant, but one should consider simply the number of experiments that it is possible to do in the course of system development. In our case, for example, it was impractical to try to try to optimize all the options and parameters of the Giza++ models in a reasonable amount of time, given the computational resources at our disposal.

While the wealth of details regarding various passes through the data in our best methods might seem to undercut our claim of simplicity, it is important to realize that each of our methods makes a fixed number of passes, and each of those passes involves a simple procedure of computing LLR scores, collecting co-occurrence counts to estimate link probabilities, or performing competitive linking; plus one best first search for minimally nonmonotonic alignments. All these procedures are simple to understand and straightforward to implement, in contrast to some of the difficult mathematical and computational issues with the standard models.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin. 2003. A Probability Model to Improve Word Alignment. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 88–95, Sapporo, Japan.

Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

William A. Gale and Kenneth W. Church. 1991. Identifying Word Correspondences in Parallel Texts. In *Proceedings of the Speech and Natural Language Workshop*, pp. 152–157, Pacific Grove, California.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pp. 127–133, Edmonton, Alberta, Canada.

I. Dan Melamed. 1998. Models of Co-occurrence. University of Pennsylvania, IRCS Technical Report #98-05.

I. Dan Melamed. 2000. Models of Translational Equivalence. *Computational Linguistics*, 26(2):221–249.

Rada Mihalcea and Ted Pedersen. 2003. An Evaluation Exercise for Word Alignment. In *Proceedings of the HLT-NAACL 2003 Workshop, Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pp. 1–6, Edmonton, Alberta, Canada.

Robert C. Moore. 2004. On Log-Likelihood-Ratios and the Significance of Rare Events. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 333–340, Barcelona, Spain.

Franz Joseph Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Cross language Text Categorization by acquiring Multilingual Domain Models from Comparable Corpora

Alfio Gliozzo and **Carlo Strapparava**
ITC-Irst
via Sommarive, I-38050, Trento, ITALY
{gliozzo,strappa}@itc.it

Abstract

In a multilingual scenario, the classical monolingual text categorization problem can be reformulated as a *cross language TC* task, in which we have to cope with two or more languages (e.g. *English* and *Italian*). In this setting, the system is trained using labeled examples in a source language (e.g. *English*), and it classifies documents in a different target language (e.g. *Italian*).

In this paper we propose a novel approach to solve the cross language text categorization problem based on acquiring Multilingual Domain Models from comparable corpora in a totally unsupervised way and without using any external knowledge source (e.g. bilingual dictionaries). These Multilingual Domain Models are exploited to define a generalized similarity function (i.e. a kernel function) among documents in different languages, which is used inside a Support Vector Machines classification framework. The results show that our approach is a feasible and cheap solution that largely outperforms a baseline.

1 Introduction

Text categorization (TC) is the task of assigning category labels to documents. Categories are usually defined according to a variety of topics (e.g. SPORT, POLITICS, etc.) and, even if a large amount of hand tagged texts is required, the state-of-the-art supervised learning techniques represent a viable and well-performing solution for monolingual categorization problems.

On the other hand in the worldwide scenario of the web age, multilinguality is a crucial issue to deal with and to investigate, leading us to reformulate most of the classical NLP problems. In particular, monolingual Text Categorization can be reformulated as a *cross language TC* task, in which we have to cope with two or more languages (e.g. *English* and *Italian*). In this setting, the system is trained using labeled examples in a source language (e.g. *English*), and it classifies documents in a different target language (e.g. *Italian*).

In this paper we propose a novel approach to solve the cross language text categorization problem based on acquiring Multilingual Domain Models (MDM) from comparable corpora in an unsupervised way. A MDM is a set of clusters formed by terms in different languages. While in the monolingual settings semantic domains are clusters of related terms that co-occur in texts regarding similar topics (Gliozzo et al., 2004), in the multilingual settings such clusters are composed by terms in different languages expressing concepts in the same semantic field. Thus, the basic relation modeled by a MDM is the domain similarity among terms in different languages. Our claim is that such a relation is sufficient to capture relevant aspects of topic similarity that can be profitably used for TC purposes.

The paper is organized as follows. After a brief discussion about comparable corpora, we introduce

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 9–16,
Ann Arbor, June 2005.

a multilingual Vector Space Model, in which documents in different languages can be represented and then compared. In Section 4 we define the MDMs and we present a totally unsupervised technique to acquire them from comparable corpora. This methodology does not require any external knowledge source (e.g. bilingual dictionaries) and it is based on Latent Semantic Analysis (LSA) (Deerwester et al., 1990). MDMs are then exploited to define a Multilingual Domain Kernel, a generalized similarity function among documents in different languages that exploits a MDM (see Section 5). The Multilingual Domain Kernel is used inside a Support Vector Machines (SVM) classification framework for TC (Joachims, 2002). In Section 6 we will evaluate our technique in a Cross Language categorization task. The results show that our approach is a feasible and cheap solution, largely outperforming a baseline. Conclusions and future works are finally reported in Section 7.

2 Comparable Corpora

Comparable corpora are collections of texts in different languages regarding similar topics (e.g. a collection of news published by agencies in the same period). More restrictive requirements are expected for parallel corpora (i.e. corpora composed by texts which are mutual translations), while the class of the multilingual corpora (i.e. collection of texts expressed in different languages without any additional requirement) is the more general. Obviously parallel corpora are also comparable, while comparable corpora are also multilingual.

In a more precise way, let $L = \{L^1, L^2, \ldots, L^l\}$ be a set of languages, let $T^i = \{t_1^i, t_2^i, \ldots, t_n^i\}$ be a collection of texts expressed in the language $L^i \in L$, and let $\psi(t_h^j, t_z^i)$ be a function that returns 1 if t_z^i is the translation of t_h^j and 0 otherwise. A *multilingual corpus* is the collection of texts defined by $T^* = \bigcup_i T^i$. If the function ψ exists for every text $t_z^i \in T^*$ and for every language L^j, and is known, then the corpus is *parallel* and *aligned* at document level.

For the purpose of this paper it is enough to assume that two corpora are comparable, i.e. they are composed by documents about the same topics and produced in the same period (e.g. possibly from different news agencies), and it is not known if a function ψ exists, even if in principle it could exist and return 1 for a strict subset of document pairs.

There exist many interesting works about using parallel corpora for multilingual applications (Melamed, 2001), such as Machine Translation, Cross language Information Retrieval (Littman et al., 1998), lexical acquisition, and so on.

However it is not always easy to find or build parallel corpora. This is the main reason because the *weaker* notion of comparable corpora is a matter recent interest in the field of Computational Linguistics (Gaussier et al., 2004).

The texts inside comparable corpora, being about the same topics (i.e. about the same semantic domains), should refer to the same concepts by using various expressions in different languages. On the other hand, most of the proper nouns, relevant entities and words that are not yet lexicalized in the language, are expressed by using their original terms. As a consequence the *same entities* will be denoted with the *same words* in different languages, allowing to automatically detect couples of translation pairs just by looking at the word shape (Koehn and Knight, 2002). Our hypothesis is that comparable corpora contain a large amount of such words, just because texts, referring to the same topics in different languages, will often adopt the same terms to denote the same entities[1].

However, the simple presence of these shared words is not enough to get significant results in TC tasks. As we will see, we need to exploit these common words to induce a second-order similarity for the other words in the lexicons.

3 The Multilingual Vector Space Model

Let $T = \{t_1, t_2, \ldots, t_n\}$ be a corpus, and $V = \{w_1, w_2, \ldots, w_k\}$ be its vocabulary. In the monolingual settings, the Vector Space Model (VSM) is a k-dimensional space $\mathbf{R}^k$, in which the text $t_j \in T$ is represented by means of the vector $\vec{t_j}$ such that the z^{th} component of $\vec{t_j}$ is the frequency of w_z in t_j. The similarity among two texts in the VSM is then estimated by computing the cosine of their vectors in the VSM.

[1] According to our assumption, a possible additional criterion to decide whether two corpora are comparable is to estimate the percentage of terms in the intersection of their vocabularies.

Unfortunately, such a model cannot be adopted in the multilingual settings, because the VSMs of different languages are mainly disjoint, and the similarity between two texts in different languages would always turn out zero. This situation is represented in Figure 1, in which both the left-bottom and the rigth-upper regions of the matrix are totally filled by zeros.

A first attempt to solve this problem is to exploit the information provided by external knowledge sources, such as bilingual dictionaries, to collapse all the rows representing translation pairs. In this setting, the similarity among texts in different languages could be estimated by exploiting the classical VSM just described. However, the main disadvantage of this approach to estimate inter-lingual text similarity is that it strongly relies on the availability of a multilingual lexical resource containing a list of translation pairs. For languages with scarce resources a bilingual dictionary could be not easily available. Secondly, an important requirement of such a resource is its coverage (i.e. the amount of possible translation pairs that are actually contained in it). Finally, another problem is that ambiguos terms could be translated in different ways, leading to collapse together rows describing terms with very different meanings.

On the other hand, the assumption of corpora comparability seen in Section 2, implies the presence of a number of common words, represented by the central rows of the matrix in Figure 1.

As we will show in Section 6, this model is rather poor because of its sparseness. In the next section, we will show how to use such words as seeds to induce a Multilingual Domain VSM, in which second order relations among terms and documents in different languages are considered to improve the similarity estimation.

4 Multilingual Domain Models

A MDM is a multilingual extension of the concept of Domain Model. In the literature, Domain Models have been introduced to represent ambiguity and variability (Gliozzo et al., 2004) and successfully exploited in many NLP applications, such us Word Sense Disambiguation (Strapparava et al., 2004), Text Categorization and Term Categorization.

A Domain Model is composed by soft clusters of terms. Each cluster represents a semantic domain, i.e. a set of terms that often co-occur in texts having similar topics. Such clusters identifies groups of words belonging to the same semantic field, and thus highly paradigmatically related. MDMs are Domain Models containing terms in more than one language.

A MDM is represented by a matrix $\mathbf{D}$, containing the degree of association among terms in all the languages and domains, as illustrated in Table 1.

	MEDICINE	COMPUTER_SCIENCE
$HIV^{e/i}$	1	0
$AIDS^{e/i}$	1	0
$virus^{e/i}$	0.5	0.5
$hospital^{e}$	1	0
$laptop^{e}$	0	1
$Microsoft^{e/i}$	0	1
$clinica^{i}$	1	0

Table 1: Example of Domain Matrix. w^e denotes English terms, w^i Italian terms and $w^{e/i}$ the common terms to both languages.

MDMs can be used to describe lexical ambiguity, variability and inter-lingual domain relations. Lexical ambiguity is represented by associating one term to more than one domain, while variability is represented by associating different terms to the same domain. For example the term *virus* is associated to both the domain COMPUTER_SCIENCE and the domain MEDICINE while the domain MEDICINE is associated to both the terms *AIDS* and *HIV*. Inter-lingual domain relations are captured by placing different terms of different languages in the same semantic field (as for example $HIV^{e/i}$, $AIDS^{e/i}$, $hospital^{e}$, and $clinica^{i}$). Most of the named entities, such as *Microsoft* and *HIV* are expressed using the same string in both languages.

When similarity among texts in different languages has to be estimated, the information contained in the MDM is crucial. For example the two sentences *"I went to the hospital to make an HIV check"* and *"Ieri ho fatto il test dell'AIDS in clinica"* (lit. *yesterday I did the AIDS test in a clinic*) are very highly related, even if they share no tokens. Having an "a priori" knowledge about the inter-lingual domain similarity among *AIDS*, *HIV*, *hospital* and *clinica* is then a useful information to

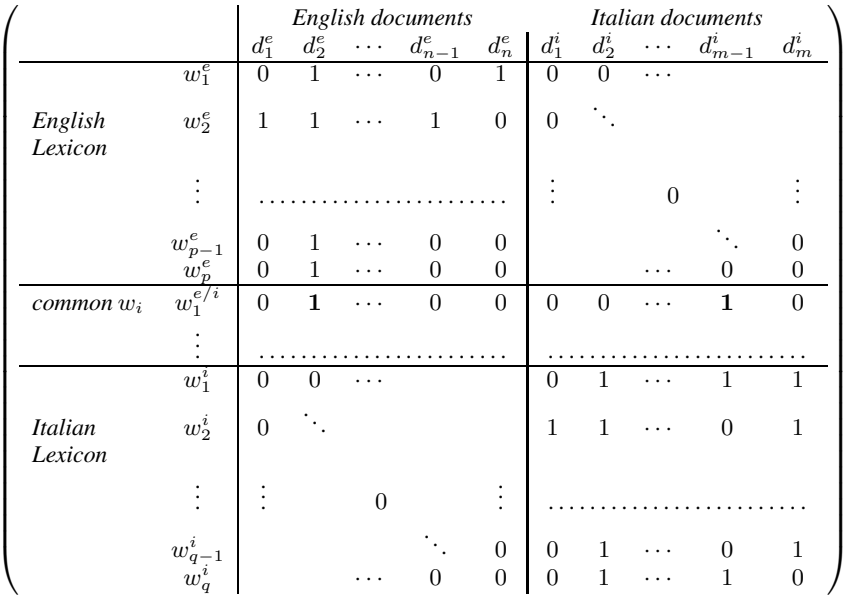


Figure 1: Multilingual term-by-document matrix

recognize inter-lingual topic similarity. Obviously this relation is less restrictive than a stronger association among translation pair. In this paper we will show that such a representation is sufficient for TC puposes, and easier to acquire.

In the rest of this section we will provide a formal definition of the concept of MDM, and we define some similarity metrics that exploit it.

Formally, let $V^i = \{w_1^i, w_2^i, \ldots, w_{k_i}^i\}$ be the vocabulary of the corpus T^i composed by document expressed in the language L^i, let $V^* = \bigcup_i V^i$ be the set of all the terms in all the languages, and let $k^* = |V^*|$ be the cardinality of this set. Let $\mathcal{D} = \{D_1, D_2, ..., D_d\}$ be a set of domains. A DM is fully defined by a $k^* \times d$ *domain matrix* $\mathbf{D}$ representing in each cell $\mathbf{d_{i,z}}$ the *domain relevance* of the i^{th} term of V^* with respect to the domain D_z. The domain matrix $\mathbf{D}$ is used to define a function $\mathcal{D} : \mathbf{R}^{k^*} \rightarrow \mathbf{R}^d$, that maps the document vectors $\vec{t_j}$ expressed into the multilingual classical VSM, into the vectors $\vec{t_j'}$ in the *multilingual domain VSM*. The function $\mathcal{D}$ is defined by[2]

$$\mathcal{D}(\vec{t_j}) = \vec{t_j}(\mathbf{I^{IDF}D}) = \vec{t_j'} \quad (1)$$

where $\mathbf{I^{IDF}}$ is a diagonal matrix such that $i_{i,i}^{IDF} = IDF(w_i^l)$, $\vec{t_j}$ is represented as a row vector, and $IDF(w_i^l)$ is the *Inverse Document Frequency* of w_i^l evaluated in the corpus T^l.

The matrix $\mathbf{D}$ can be determined for example using hand-made lexical resources, such as WORDNET DOMAINS (Magnini and Cavaglià, 2000). In the present work we followed the way to acquire $\mathbf{D}$ automatically from corpora, exploiting the technique described below.

4.1 Automatic Acquisition of Multilingual Domain Models

In this work we propose the use of Latent Semantic Analysis (LSA) (Deerwester et al., 1990) to induce a MDM from comparable corpora. LSA is an unsupervised technique for estimating the similarity among texts and terms in a large corpus. In the monolingual settings LSA is performed by means of a Singular Value Decomposition (SVD) of the term-by-document matrix $\mathbf{T}$ describing the corpus. SVD decomposes the term-by-document matrix $\mathbf{T}$ into three matrixes $\mathbf{T} \simeq \mathbf{V}\mathbf{\Sigma_{k'}}\mathbf{U}^T$ where $\mathbf{\Sigma_{k'}}$ is the diagonal $k \times k$ matrix containing the highest $k' \ll k$

[2]In (Wong et al., 1985) the formula 1 is used to define a Generalized Vector Space Model, of which the Domain VSM is a particular instance.

eigenvalues of $\mathbf{T}$, and all the remaining elements are set to 0. The parameter k' is the dimensionality of the Domain VSM and can be fixed in advance (i.e. $k' = d$).

In the literature (Littman et al., 1998) LSA has been used in multilingual settings to define a multilingual space in which texts in different languages can be represented and compared. In that work LSA strongly relied on the availability of aligned parallel corpora: documents in all the languages are represented in a term-by-document matrix (see Figure 1) and then the columns corresponding to sets of translated documents are collapsed (i.e. they are substituted by their sum) before starting the LSA process. The effect of this step is to merge the subspaces (i.e. the right and the left sectors of the matrix in Figure 1) in which the documents have been originally represented.

In this paper we propose a variation of this strategy, performing a multilingual LSA in the case in which an aligned parallel corpus is not available.

It exploits the presence of common words among different languages in the term-by-document matrix. The SVD process has the effect of creating a LSA space in which documents in both languages are represented. Of course, the higher the number of common words, the more information will be provided to the SVD algorithm to find common LSA dimension for the two languages. The resulting LSA dimensions can be perceived as multilingual clusters of terms and document. LSA can then be used to define a Multilingual Domain Matrix $\mathbf{D_{LSA}}$.

$$\mathbf{D_{LSA}} = \mathbf{I^N V}\sqrt{\mathbf{\Sigma_{k'}}} \quad (2)$$

where $\mathbf{I^N}$ is a diagonal matrix such that $\mathbf{i^N_{i,i}} = \frac{1}{\sqrt{\langle \vec{w'_i}, \vec{w'_i} \rangle}}$, $\vec{w'_i}$ is the i^{th} row of the matrix $\mathbf{V}\sqrt{\mathbf{\Sigma_{k'}}}$.

Thus $\mathbf{D_{LSA}}$[3] can be exploited to estimate similarity among texts expressed in different languages (see Section 5).

[3]When $\mathbf{D_{LSA}}$ is substituted in Equation 1 the Domain VSM is equivalent to a Latent Semantic Space (Deerwester et al., 1990). The only difference in our formulation is that the vectors representing the terms in the Domain VSM are normalized by the matrix $\mathbf{I^N}$, and then rescaled, according to their IDF value, by matrix $\mathbf{I^{IDF}}$. Note the analogy with the *tf idf* term weighting schema, widely adopted in Information Retrieval.

4.2 Similarity in the multilingual domain space

As an example of the second-order similarity provided by this approach, we can see in Table 2 the five most similar terms to the lemma *bank*. The similarity among terms is calculated by cosine among the rows in the matrix $\mathbf{D_{LSA}}$, acquired from the data set used in our experiments (see Section 6.2). It is worth noting that the Italian lemma *banca* (i.e. bank in English) has a high similarity score to the English lemma *bank*. While this is not enough to have a precise term translation, it is sufficient to capture relevant aspects of topic similarity in a cross-language text categorization task.

Lemma#Pos	*Similarity Score*	*Language*
banking#n	0.96	Eng
credit#n	0.90	Eng
amro#n	0.89	Eng
unicredito#n	0.85	Ita
banca#n	0.83	Ita

Table 2: Terms with high similarity to the English lemma *bank#n*, in the Multilingual Domain Model

5 The Multilingual Domain Kernel

Kernel Methods are the state-of-the-art supervised framework for learning, and they have been successfully adopted to approach the TC task (Joachims, 2002).

The basic idea behind kernel methods is to embed the data into a suitable feature space $\mathcal{F}$ via a mapping function $\phi : \mathcal{X} \rightarrow \mathcal{F}$, and then to use a linear algorithm for discovering nonlinear patterns. Kernel methods allow us to build a modular system, as the kernel function acts as an interface between the data and the learning algorithm. Thus the kernel function becomes the only domain specific module of the system, while the learning algorithm is a general purpose component. Potentially any kernel function can work with any kernel-based algorithm, as for example Support Vector Machines (SVMs).

During the learning phase SVMs assign a weight $\lambda_i \geq 0$ to any example $x_i \in X$. All the labeled instances x_i such that $\lambda_i > 0$ are called Support Vectors. Support Vectors lie close to the best separating hyper-plane between positive and negative examples. New examples are then assigned to the class

of the closest support vectors, according to equation 3.

$$f(x) = \sum_{i=1}^{n} \lambda_i K(x_i, x) + \lambda_0 \quad (3)$$

The kernel function $K(x_i, x)$ returns the similarity between two instances in the input space X, and can be designed just by taking care that some formal requirements are satisfied, as described in (Schölkopf and Smola, 2001).

In this section we define the Multilingual Domain Kernel, and we apply it to a cross language TC task. This kernel can be exploited to estimate the topic similarity among two texts expressed in different languages by taking into account the external knowledge provided by a MDM. It defines an explicit mapping $\mathcal{D} : \mathbf{R}^k \rightarrow \mathbf{R}^{k'}$ from the Multilingual VSM into the Multilingual Domain VSM. The Multilingual Domain Kernel is specified by

$$K_D(t_i, t_j) = \frac{\langle \mathcal{D}(t_i), \mathcal{D}(t_j) \rangle}{\sqrt{\langle \mathcal{D}(t_j), \mathcal{D}(t_j) \rangle \langle \mathcal{D}(t_i), \mathcal{D}(t_i) \rangle}} \quad (4)$$

where $\mathcal{D}$ is the Domain Mapping defined in equation 1. Thus the Multilingual Domain Kernel requires Multilingual Domain Matrix $\mathbf{D}$, in particular $\mathbf{D_{LSA}}$ that can be acquired from comparable corpora, as explained in Section 4.1.

To evaluate the Multilingual Domain Kernel we compared it to a baseline kernel function, namely the *bag_of_words* kernel, that simply estimates the topic similarity in the Multilingual VSM, as described in Section 3. The BoW kernel is a particular case of the Domain Kernel, in which $\mathbf{D} = \mathbf{I}$, and $\mathbf{I}$ is the identity matrix.

6 Evaluation

In this section we present the data set (two comparable English and Italian corpora) used in the evaluation, and we show the results of the Cross Language TC tasks. In particular we tried both to train the system on the English data set and classify Italian documents and to train using Italian and classify the English test set. We compare the learning curves of the Multilingual Domain Kernel with the standard BoW kernel, which is considered as a baseline for this task.

6.1 Implementation details

As a supervised learning device, we used the SVM implementation described in (Joachims, 1999). The Multilingual Domain Kernel is implemented by defining an explicit feature mapping as explained above, and by normalizing each vector. All the experiments have been performed with the standard SVM parameter settings.

We acquired a Multilingual Domain Model by performing the Singular Value Decomposition process on the term-by-document matrices representing the merged training partitions (i.e. English and Italian), and we considered only the first 400 dimensions[4].

6.2 Data set description

We used a news corpus kindly put at our disposal by ADNKRONOS, an important Italian news provider. The corpus consists of 32,354 Italian and 27,821 English news partitioned by ADNKRONOS in a number of four fixed categories: `Quality_of_Life`, `Made_in_Italy`, `Tourism`, `Culture_and_School`. The corpus is comparable, in the sense stated in Section 2, i.e. they covered the same topics and the same period of time. Some news are translated in the other language (but *no* alignment indication is given), some others are present only in the English set, and some others only in the Italian. The average length of the news is about 300 words. We randomly split both the English and Italian part into 75% training and 25% test (see Table 3). In both the data sets we postagged the texts and we considered only the noun, verb, adjective, and adverb parts of speech, representing them by vectors containing the frequencies of each lemma with its part of speech.

6.3 Monolingual Results

Before going to a cross-language TC task, we conducted two tests of classical monolingual TC by training and testing the system on Italian and English documents separately. For these tests we used the SVM with the BoW kernel. Figures 2 and 3 report the results.

[4]To perform the SVD operation we used LIBSVDC `http://tedlab.mit.edu/~dr/SVDLIBC/`.

Categories	*English* Training	Test	Total	*Italian* Training	Test	Total
Quality_of_Life	5759	1989	7748	5781	1901	7682
Made_in_Italy	5711	1864	7575	6111	2068	8179
Tourism	5731	1857	7588	6090	2015	8105
Culture_and_School	3665	1245	4910	6284	2104	8388
Total	20866	6955	27821	24266	8088	32354

Table 3: Number of documents in the data set partitions

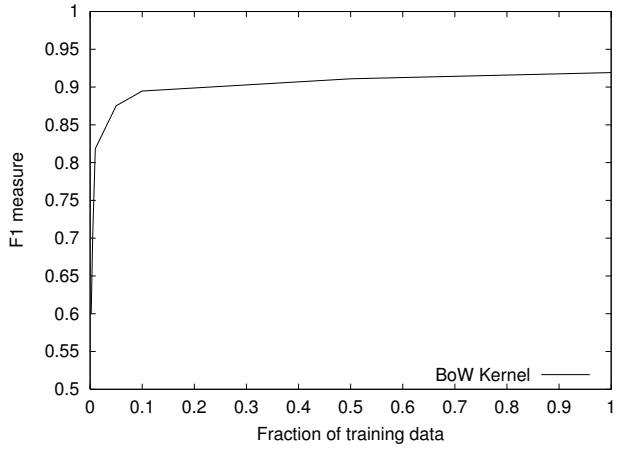


Figure 2: Learning curves for the English part of the corpus

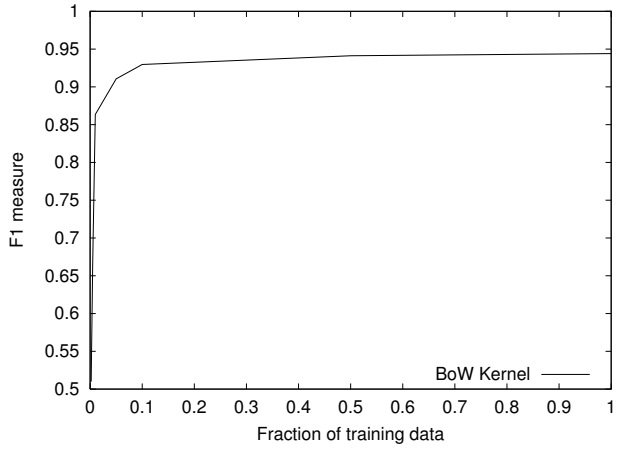


Figure 3: Learning curves for the Italian part of the corpus

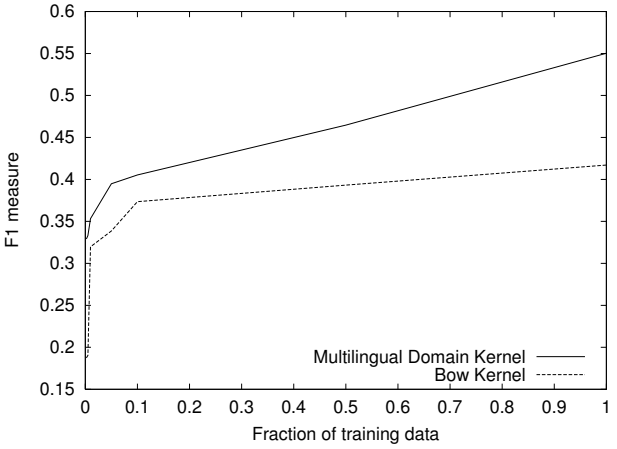


Figure 4: Cross-language (training on Italian, test on English) learning curves

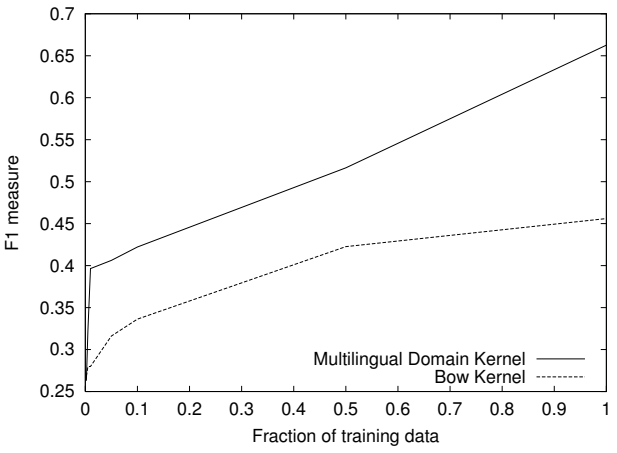


Figure 5: Cross-language (training on English, test on Italian) learning curves

6.4 A Cross Language Text Categorization task

As far as the cross language TC task is concerned, we tried the two possible options: we trained on the English part and we classified the Italian part, and we trained on the Italian and classified on the English part. The Multilingual Domain Model was acquired running the SVD only on the joint (English and Italian) training parts.

Table 4 reports the vocabulary dimensions of the English and Italian training partitions, the vocabu-

	# lemmata
English training	22,704
Italian training	26,404
English + Italian	43,384
common lemmata	5,724

Table 4: Number of lemmata in the training parts of the corpus

lary of the merged training, and how many common lemmata are present (about 14% of the total). Among the common lemmata, 97% are nouns and most of them are proper nouns. Thus the initial term-by-document matrix is a 43,384 $\times$ 45,132 matrix, while the $\mathbf{D_{LSA}}$ matrix is 43,384 $\times$ 400. For this task we consider as a baseline the BoW kernel.

The results are reported in Figures 4 and 5. Analyzing the learning curves, it is worth noting that when the quantity of training increases, the performance becomes better and better for the Multilingual Domain Kernel, suggesting that with more available training it could be possible to go closer to typical monolingual TC results.

7 Conclusion

In this paper we proposed a solution to cross language Text Categorization based on acquiring Multilingual Domain Models from comparable corpora in a totally unsupervised way and without using any external knowledge source (e.g. bilingual dictionaries). These Multilingual Domain Models are exploited to define a generalized similarity function (i.e. a kernel function) among documents in different languages, which is used inside a Support Vector Machines classification framework. The basis of the similarity function exploits the presence of common words to induce a second-order similarity for the other words in the lexicons. The results have shown that this technique is sufficient to capture relevant aspects of topic similarity in cross-language TC tasks, obtaining substantial improvements over a simple baseline. As future work we will investigate the performance of this approach to more than two languages TC task, and a possible generalization of the assumption about equality of the common words.

Acknowledgments

This work has been partially supported by the ONTOTEXT project, funded by the Autonomous Province of Trento under the FUP-2004 program.

References

S. Deerwester, S. T. Dumais, G. W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

E. Gaussier, J. M. Renders, I. Matveeva, C. Goutte, and H. Dejean. 2004. A geometric view on bilingual lexicon extraction from comparable corpora. In *Proceedings of ACL-04*, Barcelona, Spain, July.

A. Gliozzo, C. Strapparava, and I. Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18:275–299.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods: support vector learning*, chapter 11, pages 169 – 184. The MIT Press.

T. Joachims. 2002. *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers.

P. Koehn and K. Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*, Philadelphia, July.

M. Littman, S. Dumais, and T. Landauer. 1998. Automatic cross-language information retrieval using latent semantic indexing. In G. Grefenstette, editor, *Cross Language Information Retrieval*, pages 51–62. Kluwer Academic Publishers.

B. Magnini and G. Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000*, Athens, Greece, June.

D. Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts*. The MIT Press.

B. Schölkopf and A. J. Smola. 2001. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

C. Strapparava, A. Gliozzo, and C. Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation. In *Proceedings of SENSEVAL-3*, Barcelona, Spain, July.

S.K.M. Wong, W. Ziarko, and P.C.N. Wong. 1985. Generalized vector space model in information retrieval. In *Proceedings of the 8^{th} ACM SIGIR Conference*.

Parsing Word-Aligned Parallel Corpora in a Grammar Induction Context

Jonas Kuhn
The University of Texas at Austin, Department of Linguistics
jonask@mail.utexas.edu

Abstract

We present an Earley-style dynamic programming algorithm for parsing sentence pairs from a parallel corpus simultaneously, building up two phrase structure trees and a correspondence mapping between the nodes. The intended use of the algorithm is in bootstrapping grammars for less studied languages by using implicit grammatical information in parallel corpora. Therefore, we presuppose a given (statistical) word alignment underlying in the synchronous parsing task; this leads to a significant reduction of the parsing complexity. The theoretical complexity results are corroborated by a quantitative evaluation in which we ran an implementation of the algorithm on a suite of test sentences from the Europarl parallel corpus.

1 Introduction

The technical results presented in this paper[1] are motivated by the following considerations: It is conceivable to use sentence pairs from a parallel corpus (along with the tentative word correspondences from a statistical word alignment) as training data for a grammar induction approach. The goal is to induce monolingual grammars for the languages under consideration; but the implicit information about syntactic structure gathered from typical patterns in the alignment goes beyond what can be obtained from unlabeled monolingual data. Consider for instance the sentence pair from the Europarl corpus (Koehn, 2002) in fig. 1 (shown with a hand-labeled word alignment): distributional patterns over this and similar sentences may show that in English, the subject (the word block "*the situation*") is in a fixed structural position, whereas in German, it can appear in various positions; similarly, the finite verb in German (here: *stellt*) systematically appears in second position in main clauses. In a way, the translation of sentences into other natural languages serves as an approximation of a (much more costly) manual structural or semantic annotation – one might speak of automatic indirect supervision in learning. The technique will be most useful for low-resource languages and languages for which there is no funding for treebanking activities. The only requirement will be that a parallel corpus exist for the language under consideration and one or more other languages.[2]

Induction of grammars from parallel corpora is rarely viewed as a promising task in its own right; in work that has addressed the issue directly (Wu, 1997; Melamed, 2003; Melamed, 2004), the synchronous grammar is mainly viewed as instrumental in the process of improving the translation model in a noisy channel approach to statistical MT.[3] In the present paper, we provide an important prerequisite for parallel corpus-based grammar induction work: an efficient algorithm for synchronous parsing of sentence pairs, given a word alignment. This work represents a second pilot study (after (Kuhn, 2004)) for the longer-term PTOLEMAIOS project at Saarland University[4] with the goal of learning linguistic grammars from parallel corpora (compare (Kuhn, 2005)). The grammars should be robust and assign a

[1]This work was in part supported by the German Research Foundation DFG in the context of the author's Emmy Noether research group at Saarland University.

[2]In the present paper we use examples from English/German for illustration, but the approach is of course independent of the language pair under consideration.

[3]Of course, there is related work (e.g., (Hwa et al., 2002; Lü et al., 2002)) using aligned parallel corpora in order to "project" bracketings or dependency structures from English to another language and exploit them for training a parser for the other language. But note the conceptual difference: the "parse projection" approach departs from a given monolingual parser, with a particular style of analysis, whereas our project will explore to what extent it may help to design the grammar topology specifically for the parallel corpus case. This means that the emerging English parser may be different from all existing ones.

[4]http://www.coli.uni-saarland.de/~jonask/PTOLEMAIOS/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 17–24,
Ann Arbor, June 2005.

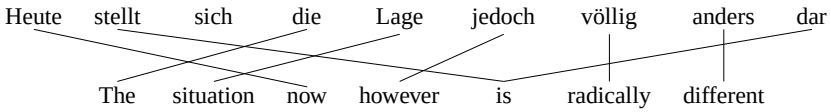


Figure 1: Word-aligned German/English sentence pair from the Europarl corpus

predicate-argument-modifier (or dependency) structure to sentences, such that they can be applied in the context of multilingual information extraction or question answering.

2 Synchronous grammars

For the purpose of grammar induction from parallel corpora, we assume a fairly straightforward extension of context-free grammars to the synchronous grammar case (compare the *transduction grammars* of (Lewis II and Stearns, 1968)): Firstly, the terminal and non-terminal categories are pairs of symbols, one for each language; as a special case, one of the two symbols can be NIL for material realized in only one of the languages. Secondly, the linear sequence of daughter categories that is specified in the rules can differ for the two languages; therefore, an explicit numerical ranking is used for the linear precedence in each language. We use a compact rule notation with a numerical ranking for the linear precedence in each language. The general form of a grammar rule for the case of two parallel languages is $N_0/M_0 \rightarrow N_1{:}i_1/M_1{:}j_1 \ldots N_k{:}i_k/M_k{:}j_k$, where N_l, M_l are NIL or a terminal or nonterminal symbol for language L_1 and L_2, respectively, and i_l, j_l are natural numbers for the rank of the phrase in the sequence for L_1 and L_2 respectively (for NIL categories a special rank 0 is assumed).[5] Since linear ordering of daughters in both languages is explicitly encoded by the rank indices, the specification sequence in the rule is irrelevant from a declarative point of view. To facilitate parsing we assume a normal form in which the right-hand side is ordered by the rank in L_1, with the exception that the categories that are NIL in L_1 come last. If there are several such NIL categories in the same rule, they are viewed as unordered with respect to each other.[6]

Fig. 2 illustrates our simple synchronous grammar formalism with some rules of a sample grammar and their application on a German/English sentence pair. Derivation with a synchronous grammar gives rise to a multitree, which combines classical phrase structure trees for the languages involved and also encodes the phrase level correspondence across the languages. Note that the two monolingual trees in fig. 2 for German and English are just two ways of unfolding the common underlying multitree.

Note that the simple formalism goes along with the **continuity assumption** that *every complete constituent is continuous in both languages*. Various recent studies in the field of syntax-based Statistical MT have shown that such an assumption is problematic when based on typical treebank-style analyses. As (Melamed, 2003) discusses for instance, in the context of binary branching structures even simple examples like the English/French pair *a gift for you from France ↔ un cadeau de France pour vouz* [*a gift from France for you*] lead to discontinuity of a "synchronous phrase" in one of the two languages. (Gildea, 2003) and (Galley et al., 2004) discuss different ways of generalizing the tree-level crosslinguistic correspondence relation, so it is not confined to single tree nodes, thereby avoiding a continuity assumption. We believe that in order to obtain full coverage on real parallel corpora, some mechanism along these lines will be required.

However, if the typical rich phrase structure analyses (with fairly detailed fine structure) are replaced by flat, multiply branching analyses, most of the highly frequent problematic cases are resolved.[7] In

[5] Note that in the probabilistic variants of these grammars, we will typically expect that *any* ordering of the right-hand side symbols is possible (but that the probability will of course vary – in a maximum entropy or log-linear model, the probability will be estimated based on a variety of learning features). This means that in parsing, the right-hand side categories will be accepted as they come in, and the relevant probability parameters are looked up accordingly.

[6] This detail will be relevant for the parsing inference rule (5) below.

[7] Compare the systematic study for English-French alignments by (Fox, 2002), who compared (i) treebank-parser style analyses, (ii) a variant with flattened VPs, and (iii) dependency structures. The degree of cross-linguistic phrasal cohesion increases from (i) to (iii). With flat clausal trees, we will come close to dependency structures with respect to cohesion.

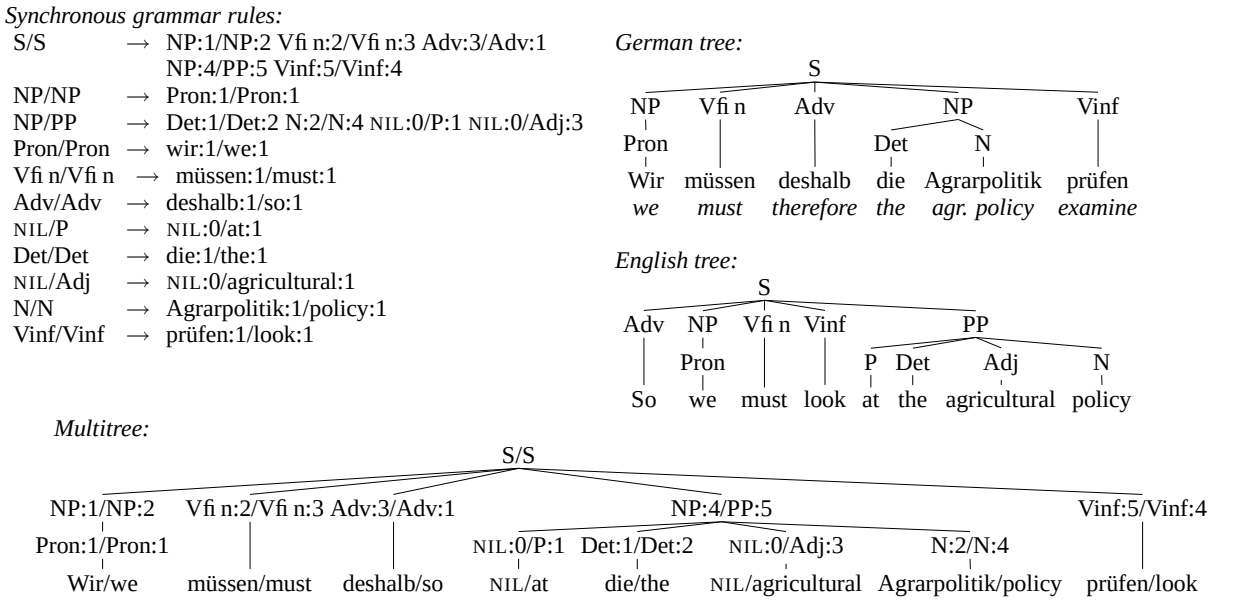


Figure 2: Sample rules and analysis for a synchronous grammar

the flat representation that we assume, a clause is represented in a single subtree of depth 1, with all verbal elements and the argument/adjunct phrases (NPs or PPs) as immediate daughters of the clause node. Similarly, argument/adjunct phrases are flat internally. Such a flat representation is justified both from the point of view of linguistic learning and from the point of view of grammar application: (i) Language-specific principles of syntactic structure (e.g., the strong configurationality of English), which are normally captured linguistically by the richer phrase structure, are available to be induced in learning as systematic patterns in the relative ordering of the elements of a clause. (ii) The predicate-argument-modifier structure relevant for application of the grammars, e.g., in information extraction can be directly read off the flat clausal representation.

It is a hypothesis of our longer-term project that a word alignment-based consensus structure which works with flat representations and under the continuity assumption is a very effective starting point for learning the basic language-specific constraints required for a syntactic grammar. Linguistic phenomena that fall outside what can be captured in this confined framework (in particular unbounded dependencies spanning more than one clause and discontinuous argument phrases) will then be learned in a later bootstrapping step that provides a richer set of operations. We are aware of a number of open practical questions, e.g.: Will the fact that real parallel corpora often contain rather free translations undermine our idea of using the consensus structure for learning basic syntactic constraints? Statistical alignments are imperfect – can the constraints imposed by the word alignment be relaxed accordingly without sacrificing tractability and the effect of indirect supervision?[8]

3 Alignment-guided synchronous parsing

Our dynamic programming algorithm can be described as a variant of standard Earley-style chart parsing (Earley, 1970) and generation (Shieber, 1988; Kay, 1996). The chart is a data structure which stores all sub-analyses that cover part of the input string (in parsing) or meaning representation (in generation). Memoizing such partial results has the standard advantage of dynamic programming techniques – it helps one to avoid unnecessary recomputation of partial results. The chart structure for context-free parsing is also exploited directly in dynamic programming algorithms for probabilistic context-free grammars (PCFGs): (i) the inside (or outside) algorithm for summing over the probabilities for every possible analysis of a given string, (ii) the Viterbi algorithm for determining the most likely analysis of a given string, and (iii) the in-

[8]Ultimately, bootstrapping of not only the grammars, but also of the word alignment should be applied.

side/outside algorithm for re-estimating the parameters of the PCFG in an Expectation-Maximization approach (i.e., for iterative training of a PCFG on unlabeled data). This aspect is important for the intended later application of our parsing algorithm in a grammar induction context.

A convenient way of describing Earley-style parsing is by inference rules. For instance, the central *completion* step in Earley parsing can be described by the rule[9]

$$(1)\quad \frac{\langle X \rightarrow \alpha \bullet Y\ \beta, [i,j]\rangle,\ \langle Y \rightarrow \gamma \bullet, [j,k]\rangle}{\langle X \rightarrow \alpha\ Y \bullet \beta, [i,k]\rangle}$$

Synchronous parsing. The input in synchronous parsing is not a one-dimensional string, but a pair of sentences, i.e., a two-dimensional array of possible word pairs (or a multidimensional array if we are looking at a multilingual corpus), as illustrated in fig. 3.

policy					•	
agricultural						
the				•		
at						
look						•
must		•				
we	•					
So			•			
	0	1	2	3	4	5 6
L_2: / L_1:	Wir	müssen	deshalb	die	Agrar-politik	prüfen

Figure 3: Synchronous parsing: two-dimensional input (with word alignment marked)

The natural way of generalizing context-free parsing to synchronous grammars is thus to control the inference rules by string indices in both dimensions. Graphically speaking, parsing amounts to identifying rectangular crosslinguistic constituents – by assembling smaller rectangles that will together cover the full string spans in both dimensions (compare (Wu, 1997; Melamed, 2003)). For instance in fig. 4, the NP/NP rectangle $[i_1, j_1, j_2, k_2]$ can be combined with the Vinf/Vinf rectangle $[j_1, k_1, i_2, j_2]$ (assuming there is an appropriate rule in the grammar).

[9]A chart item is specified through a position ($\bullet$) in a production and a string span ($[l_1, l_2]$). $\langle X \rightarrow \alpha \bullet Y\beta, [i,j]\rangle$ means that between string position i and j, the beginning of an X phrase has been found, covering α, but still missing $Y\beta$. Chart items for which the dot is at the end of a production (like $\langle Y \rightarrow \gamma\bullet, [j,k]\rangle$) are called passive items, the others active.

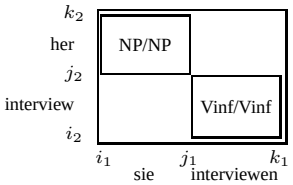


Figure 4: Completion in two-dimensional chart: parsing part of *Can I interview her?/Kann ich sie interviewen?*

More generally, we get the inference rules (2) and (3) (one for the case of parallel sequencing, one for crossed order across languages).

$$(2)\quad \frac{\langle X_1/X_2 \rightarrow \alpha \bullet\ Y_1{:}r_1/Y_2{:}r_2\ \ \beta, [i_1,j_1,i_2,j_2]\rangle,\ \langle Y_1/Y_2 \rightarrow \gamma \bullet, [j_1,k_1,j_2,k_2]\rangle}{\langle X_1/X_2 \rightarrow \alpha\ Y_1{:}r_1/Y_2{:}r_2\ \bullet\ \beta, [i_1,k_1,i_2,k_2]\rangle}$$

$$(3)\quad \frac{\langle X_1/X_2 \rightarrow \alpha \bullet\ Y_1{:}r_1/Y_2{:}r_2\ \ \beta, [i_1,j_1,j_2,k_2]\rangle,\ \langle Y_1/Y_2 \rightarrow \gamma \bullet, [j_1,k_1,i_2,j_2]\rangle}{\langle X_1/X_2 \rightarrow \alpha\ Y_1{:}r_1/Y_2{:}r_2\ \bullet\ \beta, [i_1,k_1,i_2,k_2]\rangle}$$

Since each inference rule contains six free variables over string positions ($i_1, j_1, k_1, i_2, j_2, k_2$), we get a parsing complexity of order $O(n^6)$ for unlexicalized grammars (where n is the number of words in the longer of the two strings from language L_1 and L_2) (Wu, 1997; Melamed, 2003). For large-scale learning experiments this may be problematic, especially when one moves to lexicalized grammars, which involve an additional factor of n^4.[10]

As a further issue, we observe that the inference rules are insufficient for multiply branching rules, in which *partial constituents* may be discontinuous in one dimension (only complete constituents need to be continuous in both dimensions). For instance, by parsing the first two words of the German string in fig. 1 (*Heute stellt*), we should get a partial chart item for a sentence, but the English correspondents for the two words (*now* and *is*) are discontinuous, so we couldn't apply rule (2) or (3).

Correspondence-guided parsing. As an alternative to the standard "rectangular indexing" approach

[10]The assumption here (following (Melamed, 2003)) is that lexicalization is not considered as just affecting the grammar constant, but that in parsing, every terminal symbol has to be considered as the potential head of every phrase of which it is a part. Melamed demonstrates: If the number of different category symbols is taken into consideration as l, we get $O(l^2n^6)$ for unlexicalized grammars, and $O(l^6n^{10})$ for lexicalized grammars; however there are some possible optimizations.

to synchronous parsing we propose a conceptually very simple asymmetric approach. As we will show in sec. 4 and 5, this algorithm is both theoretically and practically efficient when applied to sentence pairs for which a word alignment has previously been determined. The approach is asymmetric in that one of the languages is viewed as the "master language", i.e., indexing in parsing is mainly based on this language (the "primary index" is the string span in L_1 as in monolingual parsing). The other language contributes a secondary index, which is mainly used to guide parsing in the master language – i.e., certain options are eliminated. The choice of the master language is in principle arbitrary, but for efficiency considerations it is better to pick the one that has more words without a correspondent.

A way of visualizing correspondence-guided parsing is that standard Earley *parsing* is applied to L_1, with primary indexing by string position; as the chart items are assembled, the synchronous grammar and the information from the word alignment is used to check whether the string in L_2 could be generated (essentially using chart-based *generation* techniques; cf. (Shieber, 1988; Neumann, 1998)). The index for chart items consists of two components: the string span in L_1 and a bit vector for the words in L_2 which are covered. For instance, based on fig. 3, the noun compound *Agrarpolitik* corresponding to *agricultural policy* in English will have the index $\langle [4,5], [0,0,0,0,0,0,1,1] \rangle$ (assuming for illustrative purposes that German is the master language in this case).

The completion step in correspondence-guided parsing can be formulated as the following single inference rule:[11]

(4) $$\frac{\langle X_1/X_2 \rightarrow \alpha \bullet\ Y_1{:}r_1/Y_2{:}r_2\ \ \beta, \langle [i,j], \mathbf{v} \rangle\rangle,\ \langle Y_1/Y_2 \rightarrow \gamma \bullet, \langle [j,k], \mathbf{w} \rangle\rangle}{\langle X_1/X_2 \rightarrow \alpha\ Y_1{:}r_1/Y_2{:}r_2 \bullet\ \beta, \langle [i,k], \mathbf{u} \rangle\rangle}$$

where

(i) $j \neq k$;
(ii) $\text{OR}(\mathbf{v}, \mathbf{w}) = \mathbf{u}$;
(iii) $\mathbf{w}$ is continuous (i.e., it contains maximally one subsequence of 1's).

Condition (iii) excludes discontinuity in passive chart items, i.e., complete constituents; active items (i.e., partial constituents) may well contain discontinuities. The success condition for parsing a string with N words in L_1 is that a chart item with index $\langle [0,N], \mathbf{1} \rangle$ has been found for the start category pair of the grammar.

Words in L_2 with no correspondent in L_1 (let's call them "L_1-NIL"s for short), for example the words *at* and *agricultural* in fig. 3,[12] can in principle appear between any two words of L_1. Therefore they are represented with a "variable" empty L_1-string span like for instance in $\langle [i,i], [0,0,1,0,0] \rangle$. At first blush, such L_1-NILs seem to introduce an extreme amount of non-determinism into the algorithm. Note however that due to the continuity assumption for complete constituents, the distribution of the L_1-NILs is constrained by the other words in L_2. This is exploited by the following inference rule, which is the only way of integrating L_1-NILs into the chart:

(5) $$\frac{\langle X_1/X_2 \rightarrow \alpha \bullet \text{NIL}{:}0/Y_2{:}r_2\ \ \beta, \langle [i,j], \mathbf{v} \rangle\rangle,\ \langle \text{NIL}/Y_2 \rightarrow \gamma \bullet, \langle [j,j], \mathbf{w} \rangle\rangle}{\langle X_1/X_2 \rightarrow \alpha\ \text{NIL}{:}0/Y_2{:}r_2 \bullet\ \beta, \langle [i,j], \mathbf{u} \rangle\rangle}$$

where

(i) $\mathbf{w}$ is adjacent to $\mathbf{v}$ (i.e., unioning vectors $\mathbf{w}$ and $\mathbf{v}$ does not lead to more 0-separated 1-sequences than $\mathbf{v}$ contains already);
(ii) $\text{OR}(\mathbf{v}, \mathbf{w}) = \mathbf{u}$.

The rule has the effect of finalizing a cross-linguistic constituent (i.e., rectangle in the two-dimensional array) after all the parts that have correspondents in both languages have been found. [13]

4 Complexity

We assume that the two-dimensional chart is initialized with the correspondences following from a word alignment. Hence, for each terminal that is non-empty in L_1, both components of the index are known. When two items with known secondary indices are combined with rule (4), the new secondary

[11] We use the bold-faced variables $\mathbf{v}, \mathbf{w}, \mathbf{u}$ for bit vectors; the function OR performs bitwise disjunction on the vectors (e.g., $\text{OR}([0,1,1,0,0],[0,0,1,0,1]) = [0,1,1,0,1]$).

[12] It is conceivable that a word alignment would list *agricultural* as an additional correspondent for *Agrarpolitik*; but we use the given alignment for illustrative purposes.

[13] For instance, the L_1-NILs in fig. 3 – NIL/*at* and NIL/*agricultural* – have to be added to incomplete NP/PP constituent in the L_1-string span from 3 to 5, consisting of the Det/Det *die*/*the* and the N/N *Agrarpolitik*/*policy*. With two applications of rule (5), the two L_1-NILs can be added. Note that the conditions are met, and that as a result, we will have a continuous NP/PP constituent with index $\langle [3,5], [0,0,0,0,1,1,1,1] \rangle$, which can be used as a passive item Y_1/Y_2 in rule (4).

index can be determined by bitwise disjunction of the bit vectors. This operation is linear in the length of the L_2-string (which is of the same order as the length of the L_1-string) and has a very small constant factor.[14] Since parsing with a simple, non-lexicalized context-free grammar has a time complexity of $O(n^3)$ (due to the three free variables for string positions in the completion rule), we get $O(n^4)$ for *synchronous parsing of sentence pairs without any* L_1-NILs. Note that words from L_1 without a correspondent in L_2 (which we would have to call L_2-NILs) do not add to the complexity, so the language with more correspondent-less words can be selected as L_1.

For the *average complexity* of correspondence-guided parsing of sentence pairs without L_1-NILs we note an advantage over monolingual parsing: certain hypotheses for complete constituents that would have to be considered when parsing only L_1, are excluded because the secondary index reveals a discontinuity. An example from fig. 3 would be the sequence *müssen deshalb*, which is adjacent in L_1, but doesn't go through as a continuous rectangle when L_2 is taken into consideration (hence it cannot be used as a passive item in rule (4)).

The complexity of correspondence-guided parsing is certainly increased by the presence of L_1-NILs, since with them the secondary index can no longer be uniquely determined. However, with the adjacency condition ((i) in rule (5)), the number of possible variants in the secondary index is a function of the number of L_1-NILs. Let us say there are m L_1-NILs, i.e., the bit vectors contain m elements that we have to flip from 0 to 1 to obtain the final bit vector. In each application of rule (5) we pick a vector **v**, with a variable for the leftmost and rightmost L_1-NIL element (since this is not fully determined by the primary index). By the adjacency condition, either the leftmost or rightmost marks the boundary for adding the additional L_1-NIL element NIL/Y_2 – hence we need only one new variable for the newly shifted boundary among the L_1-NILs. So, in addition to the n^4 expense of parsing non-nil words, we get an expense of m^3 for parsing the L_1-NILs, and we conclude that for unlexicalized synchronous parsing, guided by an initial word alignment the complexity class is $O(n^4m^3)$ (where n is the total number of words appearing in L_1, and m is the number of words appearing in L_2, without a correspondent in L_1). Recall that the complexity for standard synchronous parsing is $O(n^6)$.

Since typically the number of correspondent-less words is significantly lower than the total number of words (at least for one of the two languages), these results are encouraging for medium-to-large-scale grammar learning experiments using a synchronous parsing algorithm.

5 Empirical Evaluation

In order to validate the theoretical complexity results empirically, we implemented the algorithm and ran it on sentence pairs from the Europarl parallel corpus. At the present stage, we are interested in quantitative results on parsing time, rather than qualitative results of parsing accuracy (for which a more extensive training of the rule parameters would be required).

Implementation. We did a prototype implementation of the correspondence-guided parsing algorithm in SWI Prolog.[15] Chart items are asserted to the knowledge base and efficiently retrieved using indexing by a hash function. Besides chart construction, the Viterbi algorithm for selecting the most probable analysis has been implemented, but for the current quantitative results only chart construction was relevant.

Sample grammar extraction. The initial probablistic grammar for our experiments was extracted from a small "multitree bank" of 140 German/English sentence pairs (short examples from the Europarl corpus). The multitree bank was annotated using the MMAX2 tool[16] and a specially

[14] Note that the operation does not have to be repeated when the completion rule is applied on additional pairs of items with identical indices. This means that the extra time complexity factor of n doesn't go along with an additional factor of the grammar constant (which we are otherwise ignoring in the present considerations). In practical terms this means that changes in the size of the grammar are much more noticable than moving from monolingual parsing to alignment-guided parsing.

An additional advantage is that in an Expectation Maximization approach to grammar induction (with a fixed word alignment), the bit vectors have to be computed only in the first iteration of parsing the training corpus, later iterations are cubic.

[15] http://www.swi-prolog.org – The advantage of using Prolog is that it is very easy to experiment with various conditions on the inference rules in parsing.

[16] http://mmax.eml-research.de

tailored annotation scheme for flat correspondence structures as described in sec. 2. A German and English part-of-speech tagger was used to determine word categories; they were mapped to a reduced category set and projected to the syntactic constituents.

To obtain parameters for a probabilistic grammar, we used maximum likelihood estimation from the small corpus, based on a rather simplistic generative model,[17] which for each local subtree decides (i) what categories will be the two heads, (ii) how many daughters there will be, and for each non-head sister (iii) whether it will be a nonterminal or a terminal (and in that case, what category pair), and (iv) in which position relative to the head to place it in both languages. In order to obtain a realistically-sized grammar, we applied smoothing to all parameters; so effectively, every sequence of terminals/nonterminals of arbitrary length was possible in parsing.

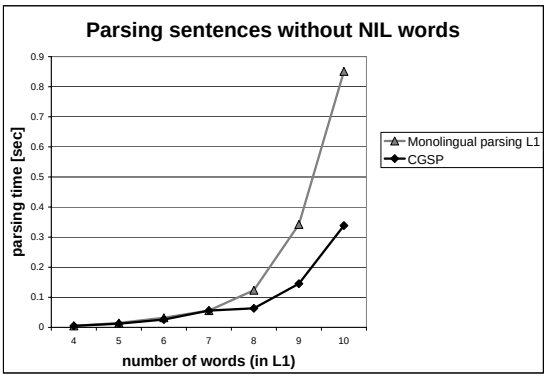


Figure 5: Comparison of synchronous parsing with and without exploiting constraints from L_2

Results. To validate empirically that the proposed correspondence-guided synchronous parsing approach (CGSP) can effectively exploit L_2 as a guide, thereby reducing the search space of L_1 parses that have to be considered, we first ran a comparison on sentences without L_1-NILs. The results (average parsing time for Viterbi parsing with the sample grammar) are shown in fig. 5.[18] The parser we call "monolingual" cannot exploit any alignment-induced restrictions from L_2.[19] Note that CGSP takes clearly less time.

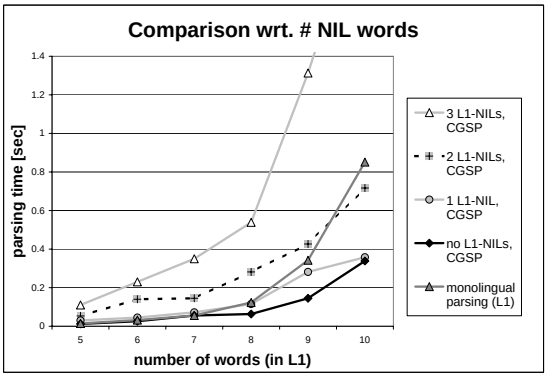


Figure 6: Synchronous parsing with a growing number of L_1-NILs

Fig. 6 shows our comparative results for parsing performance on sentences that do contain L_1-NILs. Here too, the theoretical results are corroborated that with a limited number of L_1-NILs, the CGSP is still efficient.

The average chart size (in terms of the number of entries) for sentences of length 8 (in L_1) was 212 for CGSP (and 80 for "monolingual" parsing). The following comparison shows the effect of L_1-NILs (note that the values for 4 and more L_1-NILs are based on only one or two cases):

(6) *Chart size for sentences of length 8 (in L_1)*

Number of L_1-NILs	0	1	2	3	4	5	6
Avg. number of chart items	77	121	175	256	(330)	(435)	(849)

We also simulated a synchronous parser which does not take advantage of a given word alignment (by providing an alignment link between any pair of words, plus the option that any word could be a NULL word). For sentences of length 5, this parser took an average time of 22.3 seconds (largely independent of the presence/absence of L_1-NILs).[20]

[17] For our learning experiments we intend to use a Maximum Entropy/log-linear model with more features.

[18] The experiments were run on a 1.4GHz Pentium M processor.

[19] The "monolingual" parser used in this comparison parses two identical copies of the same string synchronously, with a strictly linear alignment.

[20] While our simulation may be signifi cantly slower than a direct implementation of the algorithm (especially when some of the optimizations discussed in (Melamed, 2003) are taken into account), the fact that it is orders of magnitude slower does in-

Finally, we also ran an experiment in which the continuity condition (condition (iii) in rule (4)) was deactivated, i.e., complete constituents were allowed to be discontinuous in one of the languages. The results in (7) underscore the importance of this condition – leaving it out leads to a tremendous increase in parsing time.

(7) *Average parsing time in seconds with and without continuity condition*

Sentence length (with no L_1-NILs)	4	5	6
Avg. parsing time with CGSP (incl. continuity condition)	0.005	0.012	0.026
Avg. parsing time without the continuity condition	0.035	0.178	1.025

6 Conclusion

We proposed a conceptually simple, yet efficient algorithm for synchronous parsing in a context where a word alignment can be assumed as given – for instance in a bootstrapping learning scenario. One of the two languages in synchronous parsing acts as the master language, providing the primary string span index, which is used as in classical Earley parsing. The second language contributes a bit vector as a secondary index, inspired by work on chart generation. Continuity assumptions make it possible to constrain the search space significantly, to the point that synchronous parsing for sentence pairs with few "NULL words" (which lack correspondents) may be faster than standard monolingual parsing. We discussed the complexity both theoretically and provided a quantitative evaluation based on a prototype implementation.

The study we presented is part of the longer-term PTOLEMAIOS project. The next step is to apply the synchronous parsing algorithm with probabilistic synchronous grammars in grammar induction experiments on parallel corpora.

dicate that our correspondence-guided approach is a promising alternative for an application context in which a word alignment is available.

References

Jay C. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 304–311.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 273–280.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03), Sapporo, Japan*, pages 80–87.

Rebecca Hwa, Philip Resnik, and Amy Weinberg. 2002. Breaking the resource bottleneck for multilingual parsing. In *Proceedings of LREC*.

Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA*.

Philipp Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. Ms., University of Southern California.

Jonas Kuhn. 2004. Experiments in parallel-text based grammar induction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004*, pages 470–477.

Jonas Kuhn. 2005. An architecture for parallel corpus-based grammar learning. In Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder, and Petra Wagner, editors, *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn*, pages 132–144, Frankfurt am Main. Peter Lang.

Philip M. Lewis II and Richard E. Stearns. 1968. Syntax-directed transduction. *Journal of the Association of Computing Machinery*, 15(3):465–488.

Yajuan Lü, Sheng Li, Tiejun Zhao, and Muyun Yang. 2002. Learning chinese bracketing knowledge based on a bilingual language model. In *COLING 2002 - Proceedings of the 19th International Conference on Computational Linguistics*.

I. Dan Melamed. 2003. Multitext grammars and synchronous parsers. In *Proceedings of NAACL/HLT*.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004*, pages 653–660.

Günter Neumann. 1998. Interleaving natural language parsing and generation through uniform processing. *Artifical Intelligence*, 99:121–163.

Stuart Shieber. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING), Budapest*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Bilingual Word Spectral Clustering for Statistical Machine Translation

Bing Zhao[†] **Eric P. Xing**[† ‡] **Alex Waibel**[†]
[†]Language Technologies Institute
[‡]Center for Automated Learning and Discovery
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{bzhao,epxing,ahw}@cs.cmu.edu

Abstract

In this paper, a variant of a spectral clustering algorithm is proposed for bilingual word clustering. The proposed algorithm generates the two sets of clusters for both languages efficiently with high semantic correlation within monolingual clusters, and high translation quality across the clusters between two languages. Each cluster level translation is considered as a bilingual concept, which generalizes words in bilingual clusters. This scheme improves the robustness for statistical machine translation models. Two HMM-based translation models are tested to use these bilingual clusters. Improved perplexity, word alignment accuracy, and translation quality are observed in our experiments.

1 Introduction

Statistical natural language processing usually suffers from the sparse data problem. Comparing to the available monolingual data, we have much less training data especially for statistical machine translation (SMT). For example, in language modelling, there are more than 1.7 billion words corpora available: English Gigaword by (Graff, 2003). However, for machine translation tasks, there are typically less than 10 million words of training data.

Bilingual word clustering is a process of forming corresponding word clusters suitable for machine translation. Previous work from (Wang et al., 1996) showed improvements in perplexity-oriented measures using mixture-based translation lexicon (Brown et al., 1993). A later study by (Och, 1999) showed improvements on perplexity of bilingual corpus, and word translation accuracy using a template-based translation model. Both approaches are optimizing the maximum likelihood of parallel corpus, in which a data point is a sentence pair: an English sentence and its translation in another language such as French. These algorithms are essentially the same as monolingual word clusterings (Kneser and Ney, 1993)—an iterative local search. In each iteration, a two-level loop over every possible word-cluster assignment is tested for better likelihood change. This kind of approach has two drawbacks: first it is easily to get stuck in local optima; second, the clustering of English and the other language are basically two separated optimization processes, and cluster-level translation is modelled loosely. These drawbacks make their approaches generally not very effective in improving translation models.

In this paper, we propose a variant of the spectral clustering algorithm (Ng et al., 2001) for bilingual word clustering. Given parallel corpus, first, the word's bilingual context is used directly as features - for instance, each English word is represented by its bilingual word translation candidates. Second, latent eigenstructure analysis is carried out in this bilingual feature space, which leads to clusters of words with similar translations. Essentially an affinity matrix is computed using these cross-lingual features. It is then decomposed into two sub-spaces, which are meaningful for translation tasks: the left subspace corresponds to the representation of words in English vocabulary, and the right sub-space corresponds to words in French. Each eigenvector is considered as one bilingual concept, and the bilingual clusters are considered to be its realizations in two languages. Finally, a general K-means cluster-

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 25–32,
Ann Arbor, June 2005.

ing algorithm is used to find out word clusters in the two sub-spaces.

The remainder of the paper is structured as follows: in section 2, concepts of translation models are introduced together with two extended HMMs; in section 3, our proposed bilingual word clustering algorithm is explained in detail, and the related works are analyzed; in section 4, evaluation metrics are defined and the experimental results are given; in section 5, the discussions and conclusions.

2 Statistical Machine Translation

The task of translation is to translate one sentence in some source language *F* into a target language *E*. For example, given a French sentence with J words denoted as $f_1^J = f_1 f_2 ... f_J$, an SMT system automatically translates it into an English sentence with I words denoted by $e_1^I = e_1 e_2 ... e_I$. The SMT system first proposes multiple English hypotheses in its model space. Among all the hypotheses, the system selects the one with the highest conditional probability according to Bayes's decision rule:

$$\hat{e}_1^I = \arg\max_{\{e_1^I\}} P(e_1^I|f_1^J) = \arg\max_{\{e_1^I\}} P(f_1^J|e_1^I)P(e_1^I), \quad (1)$$

where $P(f_1^J|e_1^I)$ is called *translation model*, and $P(e_1^I)$ is called *language model*. The translation model is the key component, which is the focus in this paper.

2.1 HMM-based Translation Model

HMM is one of the effective translation models (Vogel et al., 1996), which is easily scalable to very large training corpus.

To model word-to-word translation, we introduce the mapping $j \rightarrow a_j$, which assigns a French word f_j in position j to a English word e_i in position $i = a_j$ denoted as e_{a_j}. Each French word f_j is an observation, and it is generated by a HMM state defined as $[e_{a_j}, a_j]$, where the alignment a_j for position j is considered to have a dependency on the previous alignment a_{j-1}. Thus the first-order HMM is defined as follows:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) P(a_j|a_{j-1}), \quad (2)$$

where $P(a_j|a_{j-1})$ is the transition probability. This model captures the assumption that words close in the source sentence are aligned to words close in the target sentence. An additional pseudo word of "NULL" is used as the beginning of English sentence for HMM to start with. The (Och and Ney, 2003) model includes other refinements such as special treatment of a jump to a Null word, and a uniform smoothing prior. The HMM with these refinements is used as our baseline. Motivated by the work in both (Och and Ney, 2000) and (Toutanova et al., 2002), we propose the two following simplest versions of extended HMMs to utilize bilingual word clusters.

2.2 Extensions to HMM with word clusters

Let F denote the cluster mapping $f_j \rightarrow \mathsf{F}(f_j)$, which assigns French word f_j to its cluster ID $F_j = \mathsf{F}(f_j)$. Similarly E maps English word e_i to its cluster ID of $E_i = \mathsf{E}(e_i)$. In this paper, we assume each word belongs to one cluster only.

With bilingual word clusters, we can extend the HMM model in Eqn. 1 in the following two ways:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) \cdot P(a_j|a_{j-1}, \mathsf{E}(e_{a_{j-1}}), \mathsf{F}(f_{j-1})), \quad (3)$$

where $\mathsf{E}(e_{a_{j-1}})$ and $\mathsf{F}(f_{j-1})$ are non overlapping word clusters $(E_{a_{j-1}}, F_{j-1})$for English and French respectively.

Another explicit way of utilizing bilingual word clusters can be considered as a two-stream HMM as follows:

$$P(f_1^J, F_1^J|e_1^I, E_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) P(F_j|E_{a_j}) P(a_j|a_{j-1}). \quad (4)$$

This model introduces the translation of bilingual word clusters directly as an extra factor to Eqn. 2. Intuitively, the role of this factor is to boost the translation probabilities for words sharing the same concept. This is a more expressive model because it models both word and the cluster level translation equivalence. Also, compared with the model in Eqn. 3, this model is easier to train, as it uses a two-dimension table instead of a four-dimension table.

However, we do not want this $P(F_j|E_{a_j})$ to dominate the HMM transition structure, and the obser-

vation probability of $P(f_j|e_{a_j})$ during the EM iterations. Thus a uniform prior $P(F_j) = 1/|F|$ is introduced as a smoothing factor for $P(F_j|E_{a_j})$:

$$P(F_j|E_{a_j}) = \lambda P(F_j|E_{a_j}) + (1-\lambda)P(F_j), \quad (5)$$

where $|F|$ is the total number of word clusters in French (we use the same number of clusters for both languages). λ can be chosen to get optimal performance on a development set. In our case, we fix it to be 0.5 in all our experiments.

3 Bilingual Word Clustering

In bilingual word clustering, the task is to build word clusters F and E to form partitions of the vocabularies of the two languages respectively. The two partitions for the vocabularies of F and E are aimed to be suitable for machine translation in the sense that the cluster/partition level translation equivalence is reliable and focused to handle data sparseness; the translation model using these clusters explains the parallel corpus $\{(f_1^J, e_1^I)\}$ better in terms of perplexity or joint likelihood.

3.1 From Monolingual to Bilingual

To infer bilingual word clusters of (F, E), one can optimize the joint probability of the parallel corpus $\{(f_1^J, e_1^I)\}$ using the clusters as follows:

$$\begin{aligned}(\hat{\mathsf{F}}, \hat{\mathsf{E}}) &= \arg\max_{(F,E)} P(f_1^J, e_1^I|F, E) \\ &= \arg\max_{(F,E)} P(e_1^I|E)P(f_1^J|e_1^I, F, E), \quad (6)\end{aligned}$$

Eqn. 6 separates the optimization process into two parts: the monolingual part for E, and the bilingual part for F given fixed E. The monolingual part is considered as a prior probability: $P(e_1^I|\mathsf{E})$, and E can be inferred using corpus bigram statistics in the following equation:

$$\begin{aligned}\hat{\mathsf{E}} &= \arg\max_{\{E\}} P(e_1^I|E) \\ &= \arg\max_{\{E\}} \prod_{i=1}^{I} P(E_i|E_{i-1})P(e_i|E_i). \quad (7)\end{aligned}$$

We need to fix the number of clusters beforehand, otherwise the optimum is reached when each word is a class of its own. There exists efficient leave-one-out style algorithm (Kneser and Ney, 1993), which can automatically determine the number of clusters.

For the bilingual part $P(f_1^J|e_1^I, F, E)$, we can slightly modify the same algorithm as in (Kneser and Ney, 1993). Given the word alignment $\{a_1^J\}$ between f_1^J and e_1^I collected from the Viterbi path in HMM-based translation model, we can infer $\hat{\mathsf{F}}$ as follows:

$$\begin{aligned}\hat{\mathsf{F}} &= \arg\max_{\{F\}} P(f_1^J|e_1^I, F, E) \\ &= \arg\max_{\{F\}} \prod_{j=1}^{J} P(F_j|E_{a_j})P(f_j|F_j). \quad (8)\end{aligned}$$

Overall, this bilingual word clustering algorithm is essentially a two-step approach. In the first step, E is inferred by optimizing the monolingual likelihood of English data, and secondly F is inferred by optimizing the bilingual part without changing E. In this way, the algorithm is easy to implement without much change from the monolingual correspondent.

This approach was shown to give the best results in (Och, 1999). We use it as our baseline to compare with.

3.2 Bilingual Word Spectral Clustering

Instead of using word alignment to bridge the parallel sentence pair, and optimize the likelihood in two separate steps, we develop an alignment-free algorithm using a variant of spectral clustering algorithm. The goal is to build high cluster-level translation quality suitable for translation modelling, and at the same time maintain high intra-cluster similarity , and low inter-cluster similarity for monolingual clusters.

3.2.1 Notations

We define the vocabulary V_F as the French vocabulary with a size of $|V_F|$; V_E as the English vocabulary with size of $|V_E|$. A co-occurrence matrix $C_{\{F,E\}}$ is built with $|V_F|$ rows and $|V_E|$ columns; each element represents the co-occurrence counts of the corresponding French word f_j and English word e_i. In this way, each French word forms a row vector with a dimension of $|V_E|$, and each dimensionality is a co-occurring English word. The elements in the vector are the co-occurrence counts. We can also

view each column as a vector for English word, and we'll have similar interpretations as above.

3.2.2 Algorithm

With $C_{\{F,E\}}$, we can infer two affinity matrixes as follows:

$$A_E = C_{\{F,E\}}^T C_{\{F,E\}}$$
$$A_F = C_{\{F,E\}} C_{\{F,E\}}^T,$$

where A_E is an $|V_E| \times |V_E|$ affinity matrix for English words, with rows and columns representing English words and each element the inner product between two English words column vectors. Correspondingly, A_F is an affinity matrix of size $|V_F| \times |V_F|$ for French words with similar definitions. Both A_E and A_F are *symmetric* and *non-negative*. Now we can compute the eigenstructure for both A_E and A_F. In fact, the eigen vectors of the two are correspondingly the right and left sub-spaces of the original co-occurrence matrix of $C_{\{F,E\}}$ respectively. This can be computed using singular value decomposition (SVD): $C_{\{F,E\}} = USV^T$, $A_E = VS^2V^T$, and $A_F = US^2U^T$, where U is the left sub-space, and V the right sub-space of the co-occurrence matrix $C_{\{F,E\}}$. S is a diagonal matrix, with the singular values ranked from large to small along the diagonal. Obviously, the left sub-space U is the eigenstructure for A_F; the right sub-space V is the eigenstructure for A_E.

By choosing the top K singular values (the square root of the eigen values for both A_E and A_F), the sub-spaces will be reduced to: $U_{|V_F| \times K}$ and $V_{|V_E| \times K}$ respectively. Based on these subspaces, we can carry out K-means or other clustering algorithms to infer word clusters for both languages. Our algorithm goes as follows:

- Initialize bilingual co-occurrence matrix $C_{\{F,E\}}$ with rows representing French words, and columns English words. C_{ji} is the co-occurrence raw counts of French word f_j and English word e_i;
- Form the affinity matrix $A_E = C_{\{F,E\}}^T C_{\{F,E\}}$ and $A_F = C_{\{F,E\}}^T C_{\{F,E\}}$. Kernels can also be applied here such as $A_E = \exp(\frac{C_{\{F,E\}} C_{\{F,E\}}^T}{\sigma^2})$ for English words. Set $A_{Eii} = 0$ and $A_{Fii} = 0$, and normalize each row to be unit length;
- Compute the eigen structure of the normalized matrix A_E, and find the k largest eigen vectors: $v_1, v_2, ..., v_k$; Similarly, find the k largest eigen vectors of A_F: $u_1, u_2, ..., u_k$;
- Stack the k eigenvectors of $v_1, v_2, ..., v_k$ in the columns of Y_E, and stack the eigenvectors $u_1, u_2, ..., u_k$ in the columns for Y_F; Normalize rows of both Y_E and Y_F to have unit length. Y_E is size of $|V_E| \times k$ and Y_F is size of $|V_F| \times k$;
- Treat each row of Y_E as a point in $R^{|V_E| \times k}$, and cluster them into K English word clusters using K-means. Treat each row of Y_F as a point in $R^{|V_F| \times k}$, and cluster them into K French word clusters.
- Finally, assign original word e_i to cluster E_k if row i of the matrix Y_E is clustered as E_k; similar assignments are for French words.

Here A_E and A_F are affinity matrixes of pair-wise inner products between the monolingual words. The more similar the two words, the larger the value. In our implementations, we did not apply a kernel function like the algorithm in (Ng et al., 2001). But the kernel function such as the exponential function mentioned above can be applied here to control how rapidly the similarity falls, using some carefully chosen scaling parameter.

3.2.3 Related Clustering Algorithms

The above algorithm is very close to the variants of a big family of the spectral clustering algorithms introduced in (Meila and Shi, 2000) and studied in (Ng et al., 2001). Spectral clustering refers to a class of techniques which rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters with high intra-cluster similarity and low inter-cluster similarity. It's shown to be computing the k-way normalized cut: $K - trY^T D^{-\frac{1}{2}} A D^{-\frac{1}{2}} Y$ for any matrix $Y \in R^{M \times N}$. A is the affinity matrix, and Y in our algorithm corresponds to the subspaces of U and V.

Experimentally, it has been observed that using more eigenvectors and directly computing a k-way partitioning usually gives better performance. In our implementations, we used the top 500 eigen vectors to construct the subspaces of U and V for K-means clustering.

3.2.4 K-means

The K-means here can be considered as a post-processing step in our proposed bilingual word clustering. For initial centroids, we first compute the *center* of the whole data set. The farthest centroid from the center is then chosen to be the first initial centroid; and after that, the other K-1 centroids are chosen one by one to well separate all the previous chosen centroids.

The stopping criterion is: if the maximal change of the clusters' centroids is less than the threshold of 1e-3 between two iterations, the clustering algorithm then stops.

4 Experiments

To test our algorithm, we applied it to the TIDES Chinese-English small data track evaluation test set. After preprocessing, such as English tokenization, Chinese word segmentation, and parallel sentence splitting, there are in total 4172 parallel sentence pairs for training. We manually labeled word alignments for 627 test sentence pairs randomly sampled from the dry-run test data in 2001, which has four human translations for each Chinese sentence. The preprocessing for the test data is different from the above, as it is designed for humans to label word alignments correctly by removing ambiguities from tokenization and word segmentation as much as possible. The data statistics are shown in Table 1.

		English	Chinese
Train	Sent. Pairs	4172	
	Words	133598	105331
	Voc Size	8359	7984
Test	Sent. Pairs	627	
	Words	25500	19726
	Voc Size	4084	4827
	Unseen Voc Size	1278	1888
	Alignment Links	14769	

Table 1: Training and Test data statistics

4.1 Building Co-occurrence Matrix

Bilingual word co-occurrence counts are collected from the training data for constructing the matrix of $C_{\{F,E\}}$. Raw counts are collected without word alignment between the parallel sentences. Practically, we can use word alignment as used in (Och, 1999). Given an initial word alignment inferred by HMM, the counts are collected from the aligned word pair. If the counts are L-1 normalized, then the co-occurrence matrix is essentially the bilingual word-to-word translation lexicon such as $P(f_j|e_{a_j})$. We can remove very small entries ($P(f|e) \leq 1e^{-7}$), so that the matrix of $C_{\{F,E\}}$ is more sparse for eigen-structure computation. The proposed algorithm is then carried out to generate the bilingual word clusters for both English and Chinese.

Figure 1 shows the ranked Eigen values for the co-occurrence matrix of $C_{\{F,E\}}$.

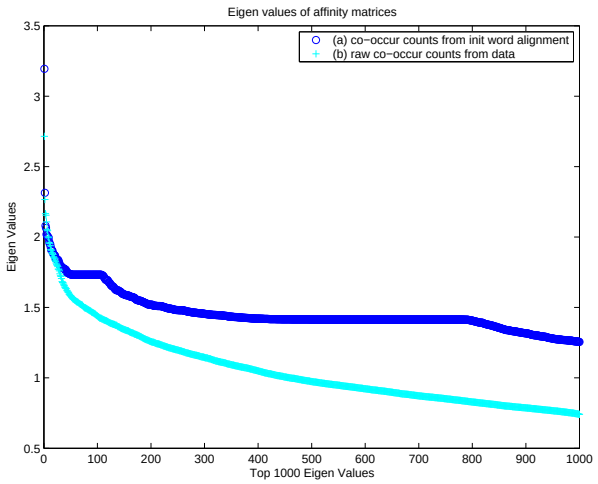


Figure 1: Top-1000 Eigen Values of Co-occurrence Matrix

It is clear, that using the initial HMM word alignment for co-occurrence matrix makes a difference. The top Eigen value using word alignment in plot a. (the deep blue curve) is 3.1946. The two plateaus indicate how many top K eigen vectors to choose to reduce the feature space. The first one indicates that K is in the range of 50 to 120, and the second plateau indicates K is in the range of 500 to 800. Plot b. is inferred from the raw co-occurrence counts with the top eigen value of 2.7148. There is no clear plateau, which indicates that the feature space is less structured than the one built with initial word alignment.

We find 500 top eigen vectors are good enough for bilingual clustering in terms of efficiency and effectiveness.

4.2 Clustering Results

Clusters built via the two described methods are compared. The first method *bil1* is the two-step optimization approach: first optimizing the monolingual clusters for target language (English), and afterwards optimizing clusters for the source language (Chinese). The second method *bil2* is our proposed algorithm to compute the eigenstructure of the co-occurrence matrix, which builds the left and right subspaces, and finds clusters in such spaces. Top 500 eigen vectors are used to construct these subspaces. For both methods, 1000 clusters are inferred for English and Chinese respectively. The number of clusters is chosen in a way that the final word alignment accuracy was optimal. Table 2 provides the clustering examples using the two algorithms.

settings	cluster examples
mono-E_1	entirely,mainly,merely
mono-E_2	10th,13th,14th,16th,17th,18th,19th 20th,21st,23rd,24th,26th
mono-E_3	drink,anglophobia,carota,giant,gymnasium
bil1-C_3	冲, 淡, 呼, 画, 啤酒, 热带, 水
bil2-E_1	alcoholic cognac distilled drink scotch spirits whiskey
bil2-C_1	白酒, 酒, 盲, 幕后, 涅, 日耳曼, 三星, 适, 苏格兰, 童, 威士忌, 蒸馏
bil2-E_2	evrec harmony luxury people sedan sedans tour tourism tourist toward travel
bil2-C_2	产业经济, 导游, 贯彻, 疾驶, 家境, 轿车, 旅行, 旅游, 人, 人民, 世人

Table 2: Bilingual Cluster Examples

The monolingual word clusters often contain words with similar syntax functions. This happens with esp. frequent words (eg. mono-E_1 and mono-E_2). The algorithm tends to put rare words such as "carota, anglophobia" into a very big cluster (eg. mono-E_3). In addition, the words within these monolingual clusters rarely share similar translations such as the typical cluster of "week, month, year". This indicates that the corresponding Chinese clusters inferred by optimizing Eqn. 7 are not close in terms of translational similarity. Overall, the method of bil1 does not give us a good translational correspondence between clusters of two languages. The English cluster of mono-E_3 and its best aligned candidate of bil1-C_3 are not well correlated either.

Our proposed bilingual cluster algorithm bil2 generates the clusters with stronger semantic meaning within a cluster. The cluster of bil2-E_1 relates to the concept of "wine" in English. The monolingual word clustering tends to scatter those words into several big noisy clusters. This cluster also has a good translational correspondent in bil2-C_1 in Chinese. The clusters of bil2-E_2 and bil2-C_2 are also correlated very well. We noticed that the Chinese clusters are slightly more noisy than their English corresponding ones. This comes from the noise in the parallel corpus, and sometimes from ambiguities of the word segmentation in the preprocessing steps.

To measure the quality of the bilingual clusters, we can use the following two kind of metrics:

- Average ϵ-mirror (Wang et al., 1996): The ϵ-mirror of a class E_i is the set of clusters in Chinese which have a translation probability greater than ϵ. In our case, ϵ is 0.05, the same value used in (Och, 1999).

- Perplexity: The perplexity is defined as proportional to the negative log likelihood of the HMM model Viterbi alignment path for each sentence pair. We use the bilingual word clusters in two extended HMM models, and measure the perplexities of the unseen test data after seven forward-backward training iterations. The two perplexities are defined as $PP1 = \exp(-\sum_{j=1}^{J} \log(P(f_j|e_{a_j})P(a_j|a_{j-1}, E_{a_{j-1}}, F_{j-1}))/J)$ and $PP2 = \exp(-J^{-1}\sum_{j=1}^{J} \log(P(f_j|e_{a_j})P(a_j|a_{j-1})P(F_{j-1}|E_{a_{j-1}})))$ for the two extended HMM models in Eqn 3 and 4.

Both metrics measure the extent to which the translation probability is spread out. The smaller the better. The following table summarizes the results on ϵ-mirror and perplexity using different methods on the unseen test data.

algorithms	ϵ-mirror	HMM-1 Perp	HMM-2 Perp
baseline	-	1717.82	
bil1	3.97	1810.55	352.28
bil2	2.54	1610.86	343.64

The baseline uses no word clusters. bil1 and bil2 are defined as above. It is clear that our proposed method gives overall lower perplexity: 1611 from the baseline of 1717 using the extended HMM-1. If we use HMM-2, the perplexity goes down even more using bilingual clusters: 352.28 using bil1, and 343.64 using bil2. As stated, the four-dimensional

table of $P(a_j|a_{j-1}, E(e_{a_{j-1}}), F(f_{j-1}))$ is easily subject to overfitting, and usually gives worse perplexities.

Average ϵ-mirror for the two-step bilingual clustering algorithm is 3.97, and for spectral clustering algorithm is 2.54. This means our proposed algorithm generates more focused clusters of translational equivalence. Figure 2 shows the histogram for the cluster pairs (F_j, E_i), of which the cluster level translation probabilities $P(F_j|E_i) \in [0.05, 1]$. The interval $[0.05, 1]$ is divided into 10 bins, with first bin $[0.05, 0.1]$, and 9 bins divides$[0.1, 1]$ equally. The percentage for clusters pairs with $P(F_j|E_i)$ falling in each bin is drawn.

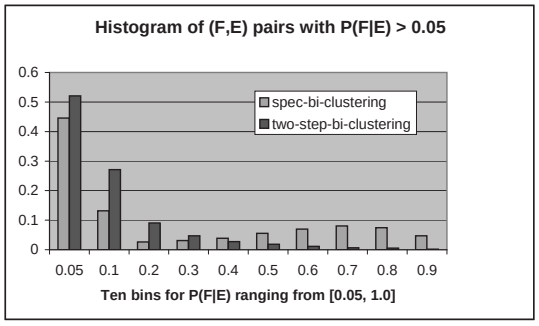


Figure 2: Histogram of cluster pairs (F_j, E_i)

Our algorithm generates much better aligned cluster pairs than the two-step optimization algorithm. There are 120 cluster pairs aligned with $P(F_j|E_i) \geq 0.9$ using clusters from our algorithm, while there are only 8 such cluster pairs using the two-step approach. Figure 3 compares the ϵ-mirror at different numbers of clusters using the two approaches. Our algorithm has a much better ϵ-mirror than the two-step approach over different number of clusters.

Overall, the extended HMM-2 is better than HMM-1 in terms of perplexity, and is easier to train.

4.3 Applications in Word Alignment

We also applied our bilingual word clustering in a word alignment setting. The training data is the TIDES small data track. The word alignments are manually labeled for 627 sentences sampled from the dryrun test data in 2001. In this manually aligned data, we include one-to-one, one-to-many, and many-to-many word alignments. Figure 4 summarizes the word alignment accuracy for different

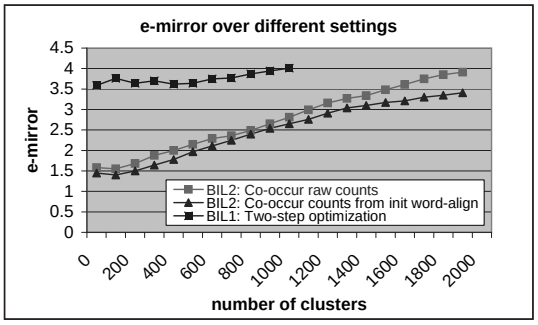


Figure 3: ϵ-mirror with different settings

methods. The baseline is the standard HMM translation model defined in Eqn. 2; the HMM1 is defined in Eqn 3, and HMM2 is defined in Eqn 4. The algorithm is applying our proposed bilingual word clustering algorithm to infer 1000 clusters for both languages. As expected, Figure 4 shows that using

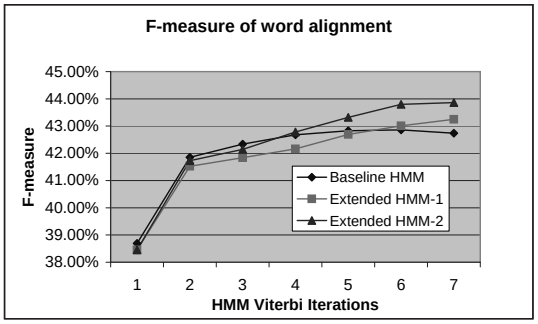


Figure 4: Word Alignment Over Iterations

word clusters is helpful for word alignment. HMM2 gives the best performance in terms of F-measure of word alignment. One quarter of the words in the test vocabulary are unseen as shown in Table 1. These unseen words related alignment links (4778 out of 14769) will be left unaligned by translation models. Thus the oracle (best possible) recall we could get is 67.65%. Our standard t-test showed that significant interval is 0.82% at the 95% confidence level. The improvement at the last iteration of HMM is marginally significant.

4.4 Applications in Phrase-based Translations

Our pilot word alignment on unseen data showed improvements. However, we find it more effective in our phrase extraction, in which three key scores

are computed: phrase level fertilities, distortions, and lexicon scores. These scores are used in a local greedy search to extract phrase pairs (Zhao and Vogel, 2005). This phrase extraction is more sensitive to the differences in $P(f_j|e_i)$ than the HMM Viterbi word aligner.

The evaluation conditions are defined in NIST 2003 Small track. Around 247K test set (919 Chinese sentences) specific phrase pairs are extracted with up to 7-gram in source phrase. A trigram language model is trained using Gigaword XinHua news part. With a monotone phrase-based decoder, the translation results are reported in Table 3. The

Eval.	Baseline	Bil1	Bil2
NIST	6.417	6.507	6.582
BLEU	0.1558	0.1575	0.1644

Table 3: NIST'03 C-E Small Data Track Evaluation

baseline is using the lexicon $P(f_j|e_i)$ trained from standard HMM in Eqn. 2, which gives a BLEU score of 0.1558 +/- 0.0113. Bil1 and Bil2 are using $P(f_j|e_i)$ from HMM in Eqn. 4 with 1000 bilingual word clusters inferred from the two-step algorithm and the proposed one respectively. Using the clusters from the two-step algorithm gives a BLEU score of 0.1575, which is close to the baseline. Using clusters from our algorithm, we observe more improvements with BLEU score of 0.1644 and a NIST score of 6.582.

5 Discussions and Conclusions

In this paper, a new approach for bilingual word clustering using eigenstructure in bilingual feature space is proposed. Eigenvectors from this feature space are considered as bilingual concepts. Bilingual clusters from the subspaces expanded by these concepts are inferred with high semantic correlations within each cluster, and high translation qualities across clusters from the two languages.

Our empirical study also showed effectiveness of using bilingual word clusters in extended HMMs for statistical machine translation. The K-means based clustering algorithm can be easily extended to do hierarchical clustering. However, extensions of translation models are needed to leverage the hierarchical clusters appropriately.

References

P.F. Brown, Stephen A. Della Pietra, Vincent. J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

David Graff. 2003. Ldc gigaword corpora: English gigaword (ldc catalog no: Ldc2003t05). In *LDC link: http://www.ldc.upenn.edu/Catalog/index.jsp*.

R. Kneser and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *European Conference on Speech Communication and Technology*, pages 973–976.

Marina Meila and Jianbo Shi. 2000. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems. (NIPS2000)*, pages 873–879.

A. Ng, M. Jordan, and Y. Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceedings of the 2001*.

Franz J. Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *COLING'00: The 18th Int. Conf. on Computational Linguistics*, pages 1086–1090, Saarbrucken, Germany, July.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29, pages 19–51.

Franz J. Och. 1999. An efficient method for determining bilingal word classes. In *Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics (EACL'99)*, pages 71–76.

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. 2002. Extensions to hmm-based statistical word alignment models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

S. Vogel, Hermann Ney, and C. Tillmann. 1996. Hmm based word alignment in statistical machine translation. In *Proc. The 16th Int. Conf. on Computational Lingustics, (Coling'96)*, pages 836–841.

Yeyi Wang, John Lafferty, and Alex Waibel. 1996. Word clustering with parallel spoken language corpora. In *proceedings of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, pages 2364–2367.

Bing Zhao and Stephan Vogel. 2005. A generalized alignment-free phrase extraction algorithm. In *ACL 2005 Workshop: Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond*, Ann Arbor, Michigan.

Revealing Phonological Similarities between Related Languages from Automatically Generated Parallel Corpora

Karin Müller
Informatics Institute
University of Amsterdam
Kruislaan 403
1098 SJ Amsterdam, The Netherlands
kmueller@science.uva.nl

Abstract

In this paper, we present an approach to automatically revealing phonological correspondences within historically related languages. We create two bilingual pronunciation dictionaries for the language pairs German-Dutch and German-English. The data is used for automatically learning phonological similarities between the two language pairs via EM-based clustering. We apply our models to predict from a phonological German word the phonemes of a Dutch and an English cognate. The similarity scores show that German and Dutch phonemes are more similar than German and English phonemes, which supplies statistical evidence of the common knowledge that German is more closely related to Dutch than to English. We assess our approach qualitatively, finding meaningful classes caused by historical sound changes. The classes can be used for language learning.

1 Introduction

German and Dutch are languages that exhibit a wide range of similarities. Beside similar syntactic features like word order and verb subcategorization frames, the languages share phonological features which are due to historical sound changes. These similarities are one reason why it is easier to learn a closely historically related language than languages from other language families: the learner's native language provides a valuable resource which can be used in learning the new language. Although English also belongs to the West Germanic languages, German and Dutch share more lexical entries with a common root than German and English.

The knowledge about language similarities on the lexical level is exploited in various fields. In machine translation, some approaches search for similar words (cognates) which are used to align parallel texts (e.g., Simard et al. (1992)). The word triple *Text-tekst-text* ([tEkst] in German, Dutch and English) can be easily recognized as a cognate; recognizing *Pfeffer-peper-pepper* ([pfE][f@r]-[pe:][p@r])-[pE][p@r*]), however, requires more knowledge about sound changes within the languages. The algorithms developed for machine translation search for similarities on the orthographic level, whereas some approaches to comparative and synchronic linguistics put their focus on similarities of phonological sequences. Covington (1996), for instance, suggests different algorithms to align the phonetic representation of words of historical languages. Kondrak (2000) presents an algorithm to align phonetic sequences by computing the similarities of these words. Nerbonne and Heeringa (1997) use phonetic transcriptions to measure the phonetic distance between different dialects. The above mentioned approaches presuppose either parallel texts of different languages for machine translation or manually compiled lists of transcribed cognates/words for analyzing synchronic or diachronic word pairs. Unfortunately, transcribed bilingual data are scarce and it

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 33–40,
Ann Arbor, June 2005.

is labor-intensive to collect these kind of corpora. Thus, we aim at exploiting electronic pronunciation dictionaries to overcome the lack of data.

In our approach, we automatically generate data as input to an unsupervised training regime and with the aim of automatically learning similar structures from these data using Expectation Maximization (EM) based clustering. Although the generation of our data introduces some noise, we expect that our method is able to automatically learn meaningful sound correspondences from a large amount of data. Our main assumption is that certain German/Dutch and German/English phoneme pairs from related stems occur more often and hence will appear in the same class with a higher probability than pairs not in related stems. We assume that the historical sound changes are hidden information in the classes.

The paper is organized as follows: Section 2 presents related research. In Section 3, we describe the creation of our bilingual pronunciation dictionaries. The outcome is used as input to the algorithm for automatically deriving phonological classes described in Section 4. In Section 5, we apply our classes to a transcribed cognate list and measure the similarity between the two language pairs. A qualitative evaluation is presented in Section 6, where we interpret our best models. In Sections 7 and 8, we discuss our results and draw some final conclusions.

2 Previous Research

Some approaches to revealing sound correspondences require clean data whereas other methods can deal with noisy input. Cahill and Tiberius (2002) use a manually compiled cognate list of Dutch, English and German cognates and extract cross-linguistic phoneme correspondences. The results[1] contain the counts of a certain German phoneme and their possible English and Dutch counterparts. The method presented in Kondrak (2003), however, can deal with noisy bilingual word lists. He generates sound correspondences of various Algonquian languages. His algorithm considers them as possible candidates if their likelihood scores lie above a certain minimum-strength threshold. The candidates are evaluated against manually compiled sound correspondences. The algorithm is able to judge whether a bilingual phoneme pair is a possible sound correspondence. Another interesting generative model can be found in Knight and Graehl (1998). They train weighted finite-state transducers with the EM algorithm which are applied to automatically transliterating Japanese words - originated from English - back to English. In our approach, we aim at discovering similar correspondences between bilingual data represented in the classes. The classes can be used to assess how likely a bilingual sound correspondence is.

[1] http://www.itri.brighton.ac.uk/projects/metaphon/

3 Generation of two parallel Corpora

In this section, we describe the resources used for our clustering algorithm. We take advantage of two on-line bilingual orthographic dictionaries[2] and the monolingual pronunciation dictionaries (Baayen et al., 1993) in CELEX to automatically build two bilingual pronunciation dictionaries.

In a first step, we extract from the German-Dutch orthographic dictionary 72,037 word pairs and from the German-English dictionary 155,317. Figures 1 and 2 (1st table) display a fragment of the extracted orthographic word pairs. Note that we only allow one possible translation, namely the first one.

In a next step, we automatically look up the pronunciation of the German, Dutch and English words in the monolingual part of CELEX. A word pair is considered for further analysis if the pronunciation of both words is found in CELEX. For instance, the first half of the word pair *Hausflur-huisgang* (corridor) does occur in the German part of CELEX but the second half is not contained within the Dutch part. Thus, this word pair is discarded. However, the words *Haus-huis-house* are found in all three monolingual pronunciation dictionaries and are used for further analysis. Note that the transcription and syllabification of the words are defined in CELEX.

The result is a list of 44,415 transcribed German-Dutch word pairs and a list of 63,297 transcribed German-English word pairs. Figures 1 and 2 (2nd table) show the result of the look-up procedure. For instance, [”haus][3]-[”hUIs] is the transcription of *Haus-huis* in the German-Dutch dictionary, while

[2] http://deatch.de/niederlande/buch.htm
http://branchenportal-deutschland.aus-stade.de/englisch-deutsch.html

[3] A syllable is transcribed within brackets ([syllable]).

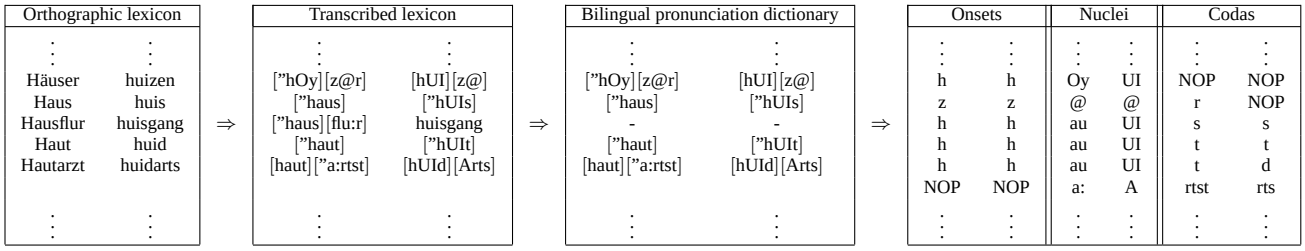


Figure 1: Creation of the **German-Dutch input**: from the orthographic lexicon - the automatically transcribed lexicon - the bilingual dictionary - to the final bilingual onset, nucleus and coda lists (left to right)

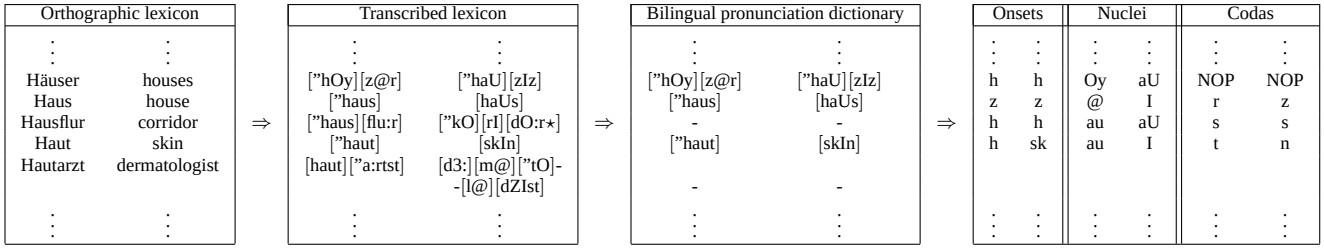


Figure 2: Creation of the **German-English input**: from the orthographic lexicon - the automatically transcribed lexicon - the bilingual dictionary - to the final bilingual onset, nucleus and coda lists (left to right)

["haus]-[haUs] is the transcription of *Haus-house* in the German-English part.

We aim at revealing phonological relationships between German-Dutch and German-English word pairs on the phonemic level, hence, we need something similar to an alignment procedure on the syllable level. Thus, we first extract only those word pairs which contain the same number of syllables. The underlying assumption is that words with a historically related stem often preserve their syllable structure. The only exception is that we do not use all inflectional paradigms of verbs to gain more data because they are often a reason for uneven syllable numbers (e.g., the past tense German suffix /tete/ is in Dutch /te/ or /de/). *Hautarzt-huidarts* would be chosen both made up of two syllables; however, *Hautarzt-dermatologist* will be dismissed as the German word consists of two syllables whereas the English word comprises five syllables. Figures 1 and 2 (3rd table) show the remaining items after this filtering process. We split each syllable within the bilingual word lists into onset, nucleus and coda. All consonants to the left of the vowel are considered the onset. The consonants to the right of the vowel represent the coda. Empty onsets and codas are replaced by the word [NOP]. After this processing step, each word pair consists of the same number of onsets, nuclei and codas.

The final step is to extract a list of German-Dutch and German-English phoneme pairs. It is easy to extract the bilingual onset, nucleus and coda pairs from the transcribed word pairs (fourth table of Figures 1 and 2). For instance, we extract the onset pair [h]-[h], the nucleus pair [au]-[UI] and the coda pair [s]-[s] from the German-Dutch word pair ["haus]-["hUIs]. With the described method, we obtain from the remaining 21,212 German-Dutch and 13,067 German-English words, 59,819 German-Dutch and 35,847 German-English onset, nucleus and coda pairs.

4 Phonological Clustering

In this section, we describe the unsupervised clustering method used for clustering of phonological units. Three- and five-dimensional EM-based clustering has been applied to monolingual phonological data (Müller et al., 2000) and two-dimensional clustering to syntax (Rooth et al., 1999). In our approach, we apply two-dimensional clustering to reveal classes of bilingual sound correspondences. The method is well-known but the application of probabilistic clustering to bilingual phonological data allows a new view on bilingual phonological

processes. We choose EM-based clustering as we need a technique which provides probabilities to deal with noise in the training data. The two main parts of EM-based clustering are (i) the induction of a smooth probability model over the data, and (ii) the automatic discovery of class structure in the data. We aim to derive a probability distribution $p(y)$ on bilingual phonological units y from a large sample ($p(c)$ denotes the class probability, $p(y_{source}|c)$ is the probability of a phoneme of the source language given class c, and $p(y_{target}|c)$ is the probability of a phoneme of the target language given class c).

$$p(y) = \sum_{c \in C} p(c) \cdot p(y_{source}|c) \cdot p(y_{target}|c)$$

The re-estimation formulas are given in (Rooth et al., 1999) and our training regime dealing with the free parameters (e.g. the number of $|c|$ of classes) is described in Sections 4.1 and 4.2. The output of our clustering algorithm are classes with their class number, class probability and a list of class members with their probabilities.

class 2	0.069	
t	0.633	t 0.764
ts	0.144	d 0.128
s	0.055	

The above table comes from our German-Dutch experiments and shows Class # 2 with its probability of 6.9%, the German onsets in the left column (e.g., [t] appears in this class with the probability of 63.3%, [ts] with 14.4% and [s] with 5.5%) and the Dutch onsets in the right column ([t] appears in this class with the probability of 76.4% and [d] with 12.8%). The examples presented in this paper are fragments of the full classes showing only those units with the highest probabilities.

4.1 Experiments with German-Dutch data

We use the 59,819 onset, nucleus and coda pairs as training material for our unsupervised training. Unsupervised methods require the variation of all free parameters to search for the optimal model. There are three different parameters which have to be varied: the initial start parameters, the number of classes and the number of re-estimation steps. Thus, we experiment with 10 different start parameters, 6 different numbers of classes (5, 10, 15, 20, 25 and 30[4]) and 20 steps of re-estimation. Our training regime yields 1,200 onset, 1,200 coda and 1,000 nucleus models.

4.2 Experiments with German-English data

Our training material is slightly smaller for German-English than for German-Dutch. We derive 35,847 onset, nucleus and coda pairs for training. The reduced training set is due to the structure of words which is less similar for German-English words than for German-Dutch words leading to words with unequal syllable numbers. We used the same training regime as in Section 4.1, yielding the same number of models.

5 Similarity scores of the syllable parts

We apply our models to a translation task. The main idea is to take a German phoneme and to predict the most probable Dutch and English counterpart.

Hence, we extract 808 German-Dutch and 738 German-English cognate pairs from a cognate database[5], consisting of 836 entries. As for the training data, we extract those pairs that consist of the same number of syllables because our current models are restricted to sound correspondences and do not allow the deletion of syllables. We split our corpus into two parts by putting the words with an even line number in the development database and the words with an uneven line number in the gold standard database. The development set and the gold standard corpus consist of 404 transcribed words for the German to Dutch translation task and of 369 transcribed words for the German to English translation task.

The task is then to predict the translation of German onsets to Dutch onsets taken from German-Dutch cognate pairs, e.g. the models should predict from the German word *durch* ([dUrx]) (through), the Dutch word *door* ([do:r]). If the phoneme correspondence, [d]:[d], is predicted, the similarity score of the onset model increases. The nucleus score increases if the nucleus model predicts [U]:[o:] and the coda score increases if the coda model predicts [rx]:[r]. We assess all our onset, nucleus and coda models

[4] We did not experiment with 30 classes for nucleus pairs as there are fewer nucleus types than onset or coda types

[5] http://www.itri.brighton.ac.uk/projects/metaphon/

German to Dutch			German to English		
Onset	Nucleus	Coda	Onset	Nucleus	Coda
80.7%	50.7 %	52.2 %	69.6%	17.1%	28.7%

Table 1: Similarity scores for syllable parts of cognates indicating that German is closer related to Dutch than to English.

by measuring the most probable phoneme translations of the cognates from our development set. We choose the models with the highest onset, nucleus and coda scores. Only the models with the highest scores (for onset, nucleus and coda prediction) are applied to the gold standard to avoid tuning to the development set. Using this procedure shows how our models perform on new data. We apply our scoring procedure to both language pairs.

Table 1 shows the results of our best models by measuring the onset, nucleus and coda translation scores on our gold standard. The results point out that the prediction of the onset is easier than predicting the nucleus or the coda. We achieve an onset similarity score of 80.7% for the German to Dutch task and 69.6% for the German to English task. Although the set of possible nuclei is smaller than the set of onsets and codas, the prediction of the nuclei is much harder. The nucleus similarity score decreases to 50.7% and to 17.1% for German-English respectively. Codas seem to be slightly easier to predict than nuclei leading to a coda similarity score of 52.2% for German-Dutch and to 28.7% for German-English.

The comparison of the similarity scores from the translation tasks of the two language pairs indicates that predicting the phonological correspondences from German to Dutch is much easier than from German to English. These results supply statistical evidence that German is historically more closely related to Dutch than to English. We do not believe that the difference in the similarity scores are due to the different size of the training corpora but rather to their closer relatedness. Revealing phonological relationships between languages is possible simply because the noisy training data comprise enough related words to learn from them the similar structure of the languages on the syllable-part level.

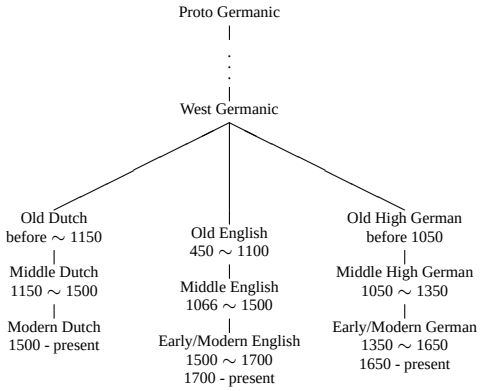


Figure 3: Family tree of West Germanic languages

6 Evaluation: Interpretation of the Classes

In this section, we interpret our classes by manually identifying classes that show typical similarities between the two language pairs. Sometimes, the classes reflect sound changes in historically related stems. Our data is synchronic, and thus it is not possible to directly identify in our classes which sound changes took place (Modern German (G), Modern English (E) and Modern Dutch (NL) did not develop from each other but from a common ancestor). However, we will try to connect the data to ancient languages such as Old High German (OHG), Middle High German (MHG), Old English (OE), Middle Dutch (MNL), Old Dutch (ONL), Proto or West Germanic (PG, WG). Naturally, we can only go back in history as far as it is possible according to the information provided by the following literature: For Dutch, we use de Vries (1997) and the online version of Philippa et al. (2004), for English, an etymological dictionary (Harper, 2001) and for German, Burch et al. (1998). We find that certain historic sound changes took place regularly, and thus, the results of these changes can be rediscovered in our synchronic classes. Figure 3 shows the historic relationship between the three languages. A potential learner of a related language does not have to be aware of the historic links between languages but he/she can implicitly exploit the similarities such as the ones discovered in the classes.

The relationship of words from different languages can be caused by different processes: some words are simply borrowed from another language and adapted to a new language. *Papagei-papegaai*

(parrot) is borrowed from Arabic and adapted to German and Dutch phonetics, where the /g/ is pronounced in German as a voiced velar plosive and in Dutch as an unvoiced velar fricative.

Other language changes are due to phonology; e.g., the Old English word [mus] (PG: muHs) was subject to diphthongization and changed to *mouse* ([maUs]) in Modern English. A similar process took place in German and Dutch, where the same word changed to the German word *Maus* (MHG: mûs) and to the Dutch word *muis* (MNL: muus). On the synchronic level, we find [au] and [aU] in the same class of a German-English model and [au] and [UI] in a German-Dutch model. There are also other phonological processes which apply to the nuclei, such as monophthongization, raising, lowering, backing and fronting. Other phonological processes can be observed in conjunction with consonants, such as assimilation, dissimilation, deletion and insertion. Some of the above mentioned phonological processes are the underlying processes of the subsequent described classes.

6.1 German-Dutch classes

According to our similarity scores presented in Section 5, the best onset model comprises 30 classes, the nucleus model 25 classes and the coda model 30 classes. We manually search for classes, which show interesting sound correspondences.

6.1.1 Onset classes

class 20	0.016	
p	0.747	
pf	0.094	
r	0.027	p 0.902
x	0.025	x 0.022
f	0.021	

The German part of class # 20 reflects Grimm's first law which states that a West Germanic [p] is often realized as a [pf] in German. The underlying phonological process is that sounds are inserted in a certain context. The onsets of the Middle High German words *phat* (E: path) and *phert* (E: horse, L: paraverēredus) became the affricate [pf] in Modern German. In contrast to German, Dutch preserved the simple onsets from the original word form, as in *paard* (E: horse, MNL: peert) and *pad* (E: path, MNL: pat).

class 25	0.012	
S	0.339	sx 0.189
Sr	0.172	sxr 0.162
ts	0.130	s 0.135
tr	0.122	tr 0.087
z	0.090	st 0.058

Class # 25 represents a class where the Dutch onsets are more complex than the onsets in German. From the Old High German word *scâf* (E: sheep) the onset /sc/ is assimilated in Modern German to [S] whereas the Dutch onset [sx] preserves the complex consonant cluster from the West Germanic word *skæpan* (E: sheep, MNL: scaep).

6.1.2 Nucleus classes

class 4	0.054	
U	0.449	O 0.721
O	0.260	U 0.112
Y	0.079	o: 0.101857
au	0.072	

We find in Class # 4 a lowering process. The German short high back vowel /U/ can be often transformed to the Dutch low back vowel /O/. The underlying processes are that the Dutch vowel is sometimes lowered from /i/ to /O/; e.g., the Dutch word *gezond* (E: healthy, MNL: ghesont, WG: gezwind) comes from the West Germanic word *gezwind*. In Modern German, the same word changed to *gesund* (OHG: gisunt).

6.1.3 Coda classes

class 14	0.027	
m	0.534	m 0.555
n	0.187	NOP 0.136
NOP	0.054	x 0.064
mt	0.042	k 0.06
mst	0.042	mt 0.055

Class # 14 represents codas where plural and infinitive suffixes /en/, as in *Menschen-mensen* (E: humans) or *laufen-lopen* (E: to run), are reduced to a Schwa [@] in Dutch and thus appear in this class with an empty coda [NOP]. It also shows that certain German codas are assimilated by the alveolar sounds /d/ and /s/ from the original bilabial [m] to an apico-alveolar [n], as in *Boden* (E: ground, MHG: bodem) or in *Besen* (E: broom, MHG: bësem, OHG: pësamo). In Dutch, the words *bodem* (E: ground, MNL: bōdem, Greek: puthmēn), and *bezem* (E: broom, MNL: bēsem, WG: besman) kept the /m/.

class 23	0.010	
rt	0.476	rt 0.521
tst	0.0782	t 0.159
rts	0.068	Nt 0.049
rst	0.067	lt 0.029
Nst	0.047	tst 0.022
t	0.023	rd 0.022
rtst	0.022	st 0.022
kt	0.021	rts 0.021
		xt 0.021

Class # 23 comprises complex German codas which are less complex in Dutch. In the German word *Arzt* (E: doctor, MHG: arzât), the complex coda [tst] emerges. However in Modern Dutch, *arts* came from MNL *arst* or *arsate* (Latin: archiāter). We can also find the rule that German codas [Nst] of a 2nd person singular form of a verb are reduced to [Nt] in Dutch as in *bringst-brengt* (E: bring).

6.2 German-English classes

The best German-English models contain 30 onset classes, 20 nucleus classes, and 10 coda classes. Our German-English models are noisier than the German-Dutch ones, which again points at the closer relation between the German and Dutch lexicon. However, when we analyze the 30 onset classes, we find meaningful processes as for German-Dutch.

6.2.1 Onset classes

class 23	0.016	
f	0.720	
Sp	0.105	
z	0.044	
S	0.012	f 0.648
v	0.011	sp 0.131
...		v 0.059
Spr	0.005	
sp	0.003	

Class # 23 shows that a complex German onset [Spr] preserves the consonant cluster, as in *sprechen* (E: to speak, OHG: sprehhan, PG: sprekanan). Modern English, however, deleted the /r/ to [sp], as in *speak* (OE: sprecan). Another regularity can be found: the palato-alveolar [S] in the German onset [Sp] is realized in English as the alveolar [s] in [sp]. Both the German word *spinnen* and the English word *spin* come from *spinnan* (OHG, OE).

class 3	0.051	
z	0.489	s 0.617
ts	0.170	z 0.143
s	0.087	

Class # 3 displays the rule that in many loan words, the onset /c/ is realized in German as [ts] and in English as [s] in *Akzent-accent* (Latin: accentus).

6.2.2 Nucleus classes

class 8	0.044	
o:	0.449	@U 0.425
y:	0.123	@ 0.201
ai	0.055	O 0.115
		u: 0.048

In some loan words, we find that an original /u/ or /o/ becomes in German the long vowel [o:] and in English the diphthong [@U], as in *Sofa-sofa* (Arabic: suffah) or in *Foto-photo* (Latin: Phosphorus). The diphthongization in English usually applies to open syllables with the nucleus /o/, as shown in class # 8.

6.2.3 Coda classes

Class # 6 displays the present participle suffix /end/, which is realized in English as /ing/ (OE: -ende), as in *backend-baking*.

class 6	0.056	
nt	0.707	N 0.846
N	0.075	NOP 0.072
lnt	0.058	nt 0.041
NOP	0.049	v 0.009
rnt	0.047	s 0.008

7 Discussion

We automatically generated two bilingual phonological corpora. The data is classified by using an EM-based clustering algorithm which is new in that respect that this method is applied to bilingual onset, nucleus and coda corpora. The method provides a probability model over bilingual syllable parts which is exploited to measure the similarity between the language pairs German-Dutch and German-English. The method is able to generalize from the data and reduces the noise introduced by the automatic generation process. Highly probable sound correspondences appear in very likely classes with a high probability whereas unlikely sound correspondences receive lower probabilities.

Our approach differs from other approaches either in the method used or in the different linguistic task. Cahill and Tiberius (2002) is based on mere counts of phoneme correspondences; Kondrak (2003) generates Algonquian phoneme correspondences which are possible according to his translation models; Kondrak (2004) measures if two words are possible cognates; and Knight and Graehl (1998) focus on the back-transliteration of Japanese words to English. Thus, we regard our approach as a thematic complement and not as an overlap to former approaches.

The presented approach depends on the available resources. That means that we can only learn those phoneme correspondences which are represented in the bilingual data. Thus, metathesis which applies to onsets and codas can not be directly observed as the syllable parts are modeled separately. In the Dutch word *borst* (ONL: bructe), the /r/ shifted from the onset to the coda whereas in English and German (*breast-Brust*), it remained in the onset. We are also

dependent on the CELEX builders, who followed different transcription strategies for the German and Dutch parts. For instance, elisions occur in the Dutch lexicon but not in the German part. The coda consonant /t/ in *lucht* (air) disappears in the Dutch word *luchtdruk* (E: air pressure), ["lUG][drUk], but not in the German word *Luftdruck*, [lUft][drUk].

We assume that the similarity scores of the syllable parts might be sharpened by increasing the size of the databases. A first possibility is to take the first transcribed translation and not the first translation in general. As often the first translation is not contained in the pronunciation dictionary.

Our current data generation process also introduces unrelated word pairs such as *Haut-skin* ([haut]-[skIn]). However, it is very unlikely that related words do not include similar phonemes. Thus, this word pair should be excluded. Exploiting this knowledge could lead to cleaner input data.

8 Conclusions and Future Work

We presented a method to automatically build bilingual pronunciation dictionaries that can be used to reveal phonological similarities between related languages. In general, our similarity scores show that the lexicons of German and Dutch are closer related than German and English. Beside the findings about the relatedness between the two language pairs, we think that the classes might be useful for language learning. An interesting point for future work is to apply the methods developed for the identification of cognates to our bilingual word-lists. Beyond the increase in data, a great challenge is to develop models that can express sound changes on the diachronic level adumbrated in Section 6. We also believe that a slightly modified version of our method can be applied to other related language pairs by using the transcription of morphemes.

9 Acknowledgments

This research was supported by the Netherlands Organization for Scientific Research under project nr. 220-80-001. I am especially endebted to B. Möbius and G. Dogil for their support and comments during a research visit in Stuttgart, as well as to D. Ahn, D. Prescher and E. Tjong Kim Sang for comments.

References

Harald R. Baayen, Richard Piepenbrock, and H. van Rijn. 1993. The CELEX lexical database—Dutch, English, German. (Release 1)[CD-ROM]. Philadelphia, PA: Linguistic Data Consortium, Univ. Pennsylvania.

Thomas Burch, Johannes Fournier, and Kurt Gärtner. 1998. Mittelhochdeutsche Wörterbücher auf CD-ROM und im Internet . *Akademie-Journal*, 2:17–24. "http://www.mwv.uni-trier.de/index.html".

Lynne Cahill and Carole Tiberius. 2002. Cross-linguistic phoneme correspondences. In *Proceedings of ACL 2002*, Taipai, Taiwan.

Michael A. Covington. 1996. An Algorithm to Align Words for Historical Comparison. *Computational Linguistics*, 22(4):481–496.

Jan de Vries. 1997. *Nederlands Etymologisch Woordenboek*. Brill, Leiden.

Daniel Harper. 2001. Online Etymology Dictionary. "http://www.etymonline.com".

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Grzegorz Kondrak. 2000. A New Algorithm for the Alignment of Phonetic Sequences. In *Proceedings of NAACL 2000*, Seattle, WA.

Grzegorz Kondrak. 2003. Identifying Complex Sound Correspondences in Bilingual Wordlists. In *Proceedings of CICLING 2003*, Mexico City.

Grzegorz Kondrak. 2004. Combining evidence in cognate identification. In *Proceedings of Canadian AI 2004*, pages 44–59.

Karin Müller, Bernd Möbius, and Detlef Prescher. 2000. Inducing Probabilistic Syllable Classes Using Multivariate Clustering. In *Proc. 38th Annual Meeting of the ACL*, Hongkong, China.

John Nerbonne and Wilbert Heeringa. 1997. Measuring Dialect Distance Phonetically. In *Proceedings of the third meeting of the SIGPHON at ACL*, pages 11–18.

Marlies Philippa, Frans Debrabandere, and Arend Quak. 2004. *Etymologisch Woordenboek van het Nederlands deel 1: A t/m E*, volume 1. Amsterdam University Press, Amsterdam. "http://www.etymologie.nl/".

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In *Proc. 37th Annual Meeting of the ACL*, College Park, MD.

Michel Simard, George F. Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of TMI-92*, Montreal Canada.

Augmenting a Small Parallel Text with Morpho-syntactic Language Resources for Serbian-English Statistical Machine Translation

Maja Popović, David Vilar, Hermann Ney
Lehrstuhl für Informatik VI
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{popovic,vilar,ney}@informatik.rwth-aachen.de

Slobodan Jovičić, Zoran Šarić
Faculty of Electrical Engineering
University of Belgrade
Serbia and Montenegro
jovicic@etf.bg.ac.yu

Abstract

In this work, we examine the quality of several statistical machine translation systems constructed on a small amount of parallel Serbian-English text. The main bilingual parallel corpus consists of about 3k sentences and 20k running words from an unrestricted domain. The translation systems are built on the full corpus as well as on a reduced corpus containing only 200 parallel sentences. A small set of about 350 short phrases from the web is used as additional bilingual knowledge. In addition, we investigate the use of monolingual morpho-syntactic knowledge i.e. base forms and POS tags.

1 Introduction and Related Work

The goal of statistical machine translation (SMT) is to translate a source language sequence $f_1, \ldots, f_J$ into a target language sequence $e_1, \ldots, e_I$ by maximising the conditional probability $Pr(e_1^I|f_1^J)$. This probability can be factorised into the translation model probability $P(f_1^J|e_1^I)$ which describes the correspondence between the words in the source and the target sequence, and the language model probability $P(e_1^I)$ which describes well-formedness of the produced target sequence. These two probabilities can be modelled independently of each other. For detailed descriptions of SMT models see for example (Brown et al., 1993; Och and Ney, 2003).

Translation probabilities are learnt from a bilingual parallel text corpus and language model probabilities are learnt from a monolingual text in the target language. Usually, the performance of a translation system strongly depends on the size of the available training corpus. However, acquisition of a large high-quality bilingual parallel text for the desired domain and language pair requires lot of time and effort, and, for many language pairs, is even not possible. Besides, small corpora have certain advantages - the acquisition does not require too much effort and also manual creation and correction are possible. Therefore there is an increasing number of publications dealing with limited amounts of bilingual data (Al-Onaizan et al., 2000; Nießen and Ney, 2004).

For the Serbian language, as a rather minor and not widely studied language, there are not many language resources available, especially not parallel texts. On the other side, investigations on this language may be quite useful since the majority of principles can be extended to the wider group of Slavic languages (e.g. Czech, Polish, Russian, etc.).

In this work, we exploit small Serbian-English parallel texts as a bilingual knowledge source for statistical machine translation. In addition, we investigate the possibilities for improving the translation quality using morpho-syntactic information in the source language. Some preliminary translation results on this language pair have been reported in (Popović et al., 2004; Popović and Ney, 2004), but no systematic investigation has been done so far. This work presents several translation systems created with different amounts and types of training data and gives a detailed description of the language resources used.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 41–48,
Ann Arbor, June 2005.

2 Language Resources

2.1 Language Characteristics

Serbian, as a Slavic language, has a very rich inflectional morphology for all open word classes. There are six distinct cases affecting not only common nouns but also proper nouns as well as pronouns, adjectives and some numbers. Some nouns and adjectives have two distinct plural forms depending on the number (if it is larger than four or not). There are also three genders for the nouns, pronouns, adjectives and some numbers leading to differences between the cases and also between the verb participles for past tense and passive voice.

As for verbs, person and many tenses are expressed by the suffix, and the subject pronoun (e.g. I, we, it) is often omitted (similarly as in Spanish and Italian). In addition, negation of three quite important verbs, "biti" (to be, auxiliary verb for past tense, conditional and passive voice), "imati" (to have) and "hteti" (to want, auxiliary verb for the future tense), is done by adding the negative particle to the verb as a prefix.

As for syntax, Serbian has a quite free word order, and there are no articles, neither indefinite nor definite.

All these characteristics indicate that morpho-syntactic knowledge might be very useful for statistical machine translation involving Serbian language, especially when only scarce amounts of parallel text are available.

2.2 Parallel Corpora

Finding high-quality bilingual or multilingual parallel corpora involving Serbian language is a difficult task. For example, there are several web-sites with the news in both Serbian and English (some of them in other languages as well), but these texts are only comparable and not parallel at all. To our knowledge, the only currently available Serbian-English parallel text suitable for statistical machine translation is a manually created electronic version of the Assimil language course which has been used for some preliminary experiments in (Popović et al., 2004; Popović and Ney, 2004). We have used this corpus for systematical investigations described in this work.

2.2.1 Assimil Language Course

The electronic form of Assimil language course contains about 3k sentences and 25k running words of various types of conversations and descriptions as well as a few short newspaper articles. Detailed corpus statistics can be seen in Table 1. Since the domain of the corpus is basically not restricted, the vocabulary size is relatively large. Due to the rich morphology, the vocabulary for Serbian is almost two times larger than for English. The average sentence length for Serbian is about 8.5 words per sentence, and for English about 9.5. This difference is mainly caused by the lack of articles and omission of some subject pronouns in Serbian .

The development and test set (500 sentences) are randomly extracted from the original corpus and the rest is used for training (referred to as 2.6k).

In order to investigate the scenario with extremely scarce training material, a reduced training corpus (referred to as 200) has been created by random extraction of 200 sentences from the original training corpus.

The morpho-syntactic annotation of the English part of the corpus has been done by the constraint grammar parser ENGCG for morphological and syntactic analysis of English language. For each word, this tool provides its base form and sequence of morpho-syntactic tags.

For the Serbian corpus, to our knowlegde there is no available tool for automatic annotation of this language. Therefore, the base forms have been introduced manually and the POS tags have been provided partly manually and partly automatically using a statistical maximum-entropy based POS tagger similar to the one described in (Ratnaparkhi, 1996). First, the 200 sentences of the reduced training corpus have been annotated completely manually. Then the first 500 sentences of the rest of the training corpus have been tagged automatically and the errors have been manually corrected. Afterwards, the POS tagger has been trained on the extended corpus (700 sentences), the next 500 sentences of the rest are annotated, and the procedure has been repeated until the annotation has been finished for the complete corpus.

Table 1: Statistics of the Serbian-English Assimil corpus

		Serbian		English	
Training:		original	base forms	original	no article
full corpus	Sentences	2632		2632	
(2.6k)	Running Words + Punct.	22227		24808	23308
	Average Sentence Length	8.4		9.5	8.8
	Vocabulary Size	4546	2605	2645	2642
	Singletons	2728	1253	1211	
reduced corpus	Sentences	200		200	
(200)	Running Words + Punct.	1666		1878	1761
	Average Sentence Length	8.3		10.4	8.8
	Vocabulary Size	778	596	603	600
	Singletons	618	417	395	
Dev+Test	Sentences	500		500	
	Running Words + Punct.	4161		4657	4362
	Average Sentence Length	8.3		9.3	8.7
	Vocabulary Size	1457	1030	1055	1052
	Running OOVs - 2.6k	12.1%	5.2%	4.8%	
	Running OOVs - 200	34.5%	27.6%	21.4%	
	OOVs - 2.6k	32.7%	19.5%	19.7%	
	OOVs - 200	76.2%	66.0%	66.8%	
External Test	Sentences	22		22	
	Running Words + Punct.	395		446	412
	Average Sentence Length	18.0		20.3	18.7
	Vocabulary Size	213	176	202	199
	Running OOVs - 2.6k	44.3%	35.4%	32.1%	34.7%
	Running OOVs - 200	53.7%	44.6%	43.7%	47.3 %
	OOVs - 2.6k	61.5%	45.4%	44.0%	44.7%
	OOVs - 200	74.6%	63.1%	63.9%	64.8%

Table 2: Statistics of the Serbian-English short phrases

		Serbian		English	
Phrases		original	base forms	original	no article
	Entries	351	351	351	351
	Running Words + Punct.	617	617	730	700
	Average Entry Length	1.8	1.8	2.1	2.0
	Vocabulary Size	335	303	315	312
	Singletons	239	209	209	208
New Running	2.6k	20.6%	14.4%	11.8%	11.8%
Words	200	50.6%	41.3%	36.7%	37.8%
New Vocabulary	2.6k	30.1%	22.1%	21.6%	21.2%
Words	200	70.7%	63.0%	63.2%	63.1%

2.2.2 Short Phrases

The short phrases used as an additional bilingual knowledge source in our experiments have been collected from the web and contain about 350 standard words and short expressions with an average entry length of 1.8 words for Serbian and 2 words for English. Table 2 shows that about 30% of words from the phrase vocabulary are not present in the original Serbian corpus and about 70% of those words are not contained in the reduced corpus. For the English language those numbers are smaller, about 20% for the original corpus and 60% for the reduced one. These percentages are indicating that this parallel text, although very scarce, might be an useful additional training material.

The phrases have also been morpho-syntactically annotated in the same way as the main corpus.

2.2.3 External Test

In addition to the standard development and test set described in Section 2.2.1, we also tested our translation systems on a short external parallel text collected from the BBC News web-site containing 22 sentences about relations between USA and Ukraine after the revolution. As can be seen in Table 1, this text contains very large portion of out-of-vocabulary words (almost two thirds of Serbian words and almost half of English words are not seen in the training corpus), and has an average sentence length about two times larger than the training corpus.

3 Transformations in the Source Language

Standard SMT systems usually regard only full forms of the words, so that translation of full forms which have not been seen in the training corpus is not possible even if the base form has been seen. Since the inflectional morphology of the Serbian language is very rich, as described in Section 2.1, we investigate the use of the base forms instead of the full forms to overcome this problem for the translation into English. We propose two types of transformations of the Serbian corpus: conversion of the full forms into the base forms and additional treatment of the verbs.

For the other translation direction, we propose removing the articles in the English part of the corpus as the Serbian language does not have any.

3.1 Transformations of the Serbian Text

3.1.1 Base Forms

Serbian full forms of the words usually contain information which is not relevant for translation into English. Therefore, we propose conversion of all Serbian words in their base forms. Although for some other inflected languages like German and Spanish this method did not yield any translation improvement, we still considered it as promising because the number of Serbian inflections is considerably higher than in the other two languages. Table 1 shows that this transformation significantly reduces the Serbian vocabulary size so that it becomes comparable to the English one.

3.1.2 Treatment of Verbs

Inflections of Serbian verbs might contain relevant information about the person, which is especially important when the pronoun is omitted. Therefore, we apply an additional treatment of the verbs. Whereas all other word classes are still replaced only by their base forms, for each verb a part of the POS tag referring to the person is taken and the verb is converted into a sequence of this tag and its base form. For the three verbs described in Section 2.1, the separation of the negative particle is also applied: each negative full form is transformed into the sequence of the POS tag, negative particle and base form. The detailed statistics of this corpus is not reported since there are no significant changes, only the number of running words and average sentence length increase thus becoming closer to the values of the English corpus.

3.2 Transformations of the English Text

3.2.1 Removing Articles

Since the articles are one of the most frequent word classes in English, but on the other side there are no arcticles at all in Serbian, we propose removing the articles from the English corpus for translation into Serbian. Each English word which has been detected as an article by means of its POS tag has been removed from the corpus. In Table 1, it can be seen that this method significantly reduces the number of running words and the average sentence length of the English corpus thus becoming comparable to the values of the Serbian corpus.

4 Translation Experiments and Results

4.1 Experimental Settings

In order to systematically investigate the impact of the bilingual training corpus size and the effects of the morpho-syntactic information on the translation quality, the translation systems were trained on the full training corpus (2.6k) and on the reduced training corpus (200), both with and without short phrases. The translation is performed in both directions, i.e. from Serbian to English and other way round. For the Serbian to English translation systems, three versions of the Serbian corpus have been used: original (baseline), base forms only (sr_base) and base forms with additional treatment of the verbs (sr_base+v-pos). For the translation into Serbian, the systems were trained on two versions of the English corpus: original (baseline) and without articles (en_no-article).

The baseline translation system is the Alignment Templates system with scaling factors (Och and Ney, 2002). Word alignments are produced using GIZA++ toolkit without symmetrisation (Och and Ney, 2003). Preprocessing of the source data has been done before the training of the system, therefore modifications of the training and search procedure were not necessary for the translation of the transformed source language corpora.

Although the development set has been used to optimise the scaling factors, results obtained for this set do not differ from those for the test set. Therefore only the joint error rates (Development+Test) are reported.

As for the external test set, results for this text are reported only for the full corpus systems, since for the reduced corpus the error rates are higher but the effects of using phrases and morpho-syntactic information are basically the same.

4.2 Translation Results

The evaluation metrics used in our experiments are WER (Word Error Rate), PER (Position-independent word Error Rate) and BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002). Since BLEU is an accuracy measure, we use 1-BLEU as an error measure.

4.2.1 Translation from Serbian into English

Error rates for the translation from Serbian into English are shown in Table 3 and some examples are shown in Table 6. It can be seen that there is a significant decrease in all error rates when the full forms are replaced with their base forms. Since the redundant information contained in the inflection is removed, the system can better capture the relevant information and is capable of producing correct or approximatively correct translations even for unseen full forms of the words (marked by "UNKNOWN_" in the baseline result example). The treatment of the verbs yields some additional improvements.

From the first translation example in Table 6 it can be seen how the problem of some out-of-vocabulary words can be overcomed with the use of the base forms. The second and third example are showing the advantages of the verb treatment, the third one illustrates the effect of separating the negative particle.

Reduction of the training corpus to only 200 sentences (about 8% of the original corpus) leads to a loss of error rates of about 45% relative. However, the degradation is not higher than 35% if phrases and morpho-syntactic information are available in addition to the reduced corpus.

The use of the phrases can improve the translation quality to some extent, especially for the systems with the reduced training corpus, but these improvements are less remarkable than those obtained by replacing words with the base forms.

The best system with the complete corpus as well as the best one with the reduced corpus use the phrases and the transformed Serbian corpus where the verb treatment has been applied.

4.2.2 Translation from English into Serbian

Table 4 shows results for the translation from English into Serbian. As expected, all error rates are higher than for the other translation direction. Translation into the morphologically richer language always has poorer quality because it is difficult to find the correct inflection.

The performance of the reduced corpus is degraded for about 40% relative for the baseline system and for about 30% when the phrases are used and the transformation of the English corpus has been applied.

Table 3: Translation error rates [%] for Serbian→English

Serbian → English		Development+Test		
Training Corpus	Method	WER	PER	1-BLEU
2.6k	baseline	45.6	39.6	70.0
2.6k	sr_base	43.5	38.2	68.9
2.6k	sr_base+v-pos	42.5	35.3	66.2
2.6k+phrases	baseline	46.0	39.6	69.5
2.6k+phrases	sr_base	44.6	39.1	70.2
2.6k+phrases	sr_base+v-pos	**42.1**	**35.3**	**66.0**
200	baseline	66.5	61.1	91.6
200	sr_base	63.2	58.2	90.3
200	sr_base+v-pos	63.3	56.2	88.5
200+phrases	baseline	65.2	59.5	90.2
200+phrases	sr_base	62.3	56.9	87.7
200+phrases	sr_base+v-pos	**61.3**	**53.2**	**86.2**

Table 4: Translation error rates [%] for English→Serbian

English → Serbian		Development+Test		
Training Corpus	Method	WER	PER	1-BLEU
2.6k	baseline	53.1	46.9	78.6
2.6k	en_no-article	52.6	47.2	79.4
2.6k+phrases	baseline	52.5	**46.5**	**76.6**
2.6k+phrases	en_no-article	**52.3**	47.0	79.6
200	baseline	73.6	68.0	93.0
200	en_no-article	71.5	66.5	93.4
200+phrases	baseline	71.7	66.7	92.3
200+phrases	en_no-article	**67.9**	**62.9**	**92.1**

Table 5: Translation error rates [%] for the external test

Serbian → English		External Test		
Training Corpus	Method	WER	PER	1-BLEU
2.6k	baseline	72.2	64.8	92.2
2.6k	sr_base	**66.8**	61.4	**86.9**
2.6k	sr_base+v-pos	67.5	61.4	88.3
2.6k+phrases	baseline	71.3	63.9	91.9
2.6k+phrases	sr_base	67.0	**61.2**	88.4
2.6k+phrases	sr_base+v-pos	69.7	**61.2**	89.8
English → Serbian				
2.6k	baseline	85.3	77.0	96.4
2.6k	en_no-article	**77.5**	**69.9**	95.8
2.6k+phrases	baseline	84.1	74.9	95.2
2.6k+phrases	en_no-article	77.7	70.1	**94.8**

The importance of the phrases seems to be larger for this translation direction. Removing the English articles does not have the significant role for the translation systems with full corpus, but for the reduced corpus it has basically the same effect as the use of phrases. The best system with the reduced corpus has been built with the use of phrases and removal of the articles.

Table 7 shows some examples of the translation into Serbian with and without English articles. Although these effects are not directly obvious, it can be seen that removing of the redundant information enables better learning of the relevant information so that system is better capable of producing semantically correct output. The first example illustrates an syntactically incorrect output with the wrong inflection of the verb ("čitam" means "I read"). The output of the system without articles is still not completely correct, but the semantic is completely preserved. The second example illustrates an output produced by the baseline system which is neither syntactically nor semantically correct ("you have I drink"). The output of the new system still has an error in the verb, informal form of "you" instead of the formal one, but nevertheless both the syntax and semantics are correct.

4.2.3 Translation of the External Text

Translation results for the *external test* can be seen in Table 5. As expected, the high number of out-of-vocabulary words results in very high error rates. Certain improvement is achieved with the phrases, but the most significant improvements are yielded by the use of Serbian base forms and removal of English articles. Verb treatment in this case does not outperform the base forms system, probably because there are not so many different verb forms as in the other corpus, and only a small number of pronouns is missing.

5 Conclusions

In this work, we have examined the possibilities for building a statistical machine translation system with a small bilingual Serbian-English parallel text. Our experiments showed that the translation results for this language pair are comparable with results for other language pairs, especially if the small size of the corpus, unrestricted domain and rich inflectional morphology of Serbian language are taken into account. With the baseline system, we obtained about 45% WER for translation into English and about 53% for translation into Serbian.

We have systematically investigated the impact of the corpus size on translation quality, as well as the importance of additional bilingual knowledge in the form of short phrases. In addition, we have shown that morpho-syntactic information is a valuable language resource for translation of this language pair.

Depending on the availability of resources and tools, we plan to examine parallel texts with other languages, and also to do further investigations on this language pair. We believe that more refined use of the morpho-syntactic information can yield better results (for example the hierarchical lexicon model proposed in (Nießen and Ney, 2001)). We also believe that the use of the conventional dictionaries could improve the Serbian-English translation.

Acknowledgement

This work was partly funded by the Deutsche Forschungsgemeinschaft (DFG) under the project "Statistical Methods for Written Language Translation" (Ne572/5).

References

Y. Al-Onaizan, U. Germann, U. Hermjakob, K. Knight, P. Koehn, D. Marcu, and K. Yamada. 2000. Translating with scarce resources. In *National Conference on Artificial Intelligence (AAAI)*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Sonja Nießen and Hermann Ney. 2001. Toward hierarchical models for statistical machine translation of inflected languages. In *39th Annual Meeting of the Assoc. for Computational Linguistics - joint with EACL 2001: Proc. Workshop on Data-Driven Machine Translation*, pages 47–54, Toulouse, France, July.

Sonja Nießen and Hermann Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181–204, June.

Franz J. Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical

Table 6: Examples of Serbian–English translations with and without transformations

to je suvishe *skupo* .	⇒ base forms	to biti suvishe *skup* .	⇒ verb treatment	to SG3 biti suvishe *skup* .
⇓ Sr → En (baseline)		⇓ Sr' → En		⇓ Sr" → En
it is too *UNKNOWN_skupo* .		it is too *expensive* .		it is too *expensive* .
on ne *igra* .	⇒ base forms	on ne *igrati* .	⇒ verb treatment	on ne *SG3 igrati* .
⇓ Sr → En (baseline)		⇓ Sr' → En		⇓ Sr" → En
he he does not .		he *do not* play .		he *does not play* .
da , ali *nemam* mnogo vremena .	⇒ base forms	da , ali *nemati* mnogo vreme .	⇒ verb treatment	da , ali *SG1 ne imati* mnogo vreme .
⇓ Sr → En (baseline)		⇓ Sr' → En		⇓ Sr" → En
yes , but *I have* much time .		yes , but *not* much time .		yes , but *I have not got* much time .

Table 7: Examples of English–Serbian translations with and without transformations

you should not read in bed .	⇒ remove articles	you should not read in bed .	
⇓ En → Sr (baseline)		⇓ En' → Sr	reference translation:
treba ne čitam u krevet .		ne bi trebalo čitate u krevet .	ne bi trebalo da čitate u krevetu .
have a drink .	⇒ remove articles	have drink .	
⇓ En → Sr (baseline)		⇓ En' → Sr	reference translation:
imate pijem .		uzmi nešto za piće .	uzmite nešto za piće .

machine translation. In *Proc. 40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

Kishore Papineni, Salim Roukos, Todd Ward, and Wie-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. 40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, July.

M. Popović and H. Ney. 2004. Towards the use of word stems and suffixes for statistical machine translation. In *Proc. 4th Int. Conf. on Language Resources and Evaluation (LREC)*, pages 1585–1588, Lisbon, Portugal, May.

M. Popović, S. Jovičić, and Z. Šarić. 2004. Statistical machine translation of Serbian-English. In *Proc. of Int. Workshop on Speech and Computer (SPECOM)*, pages 410–414, St. Petersburg, Russia, September.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 133–142, Sommerset, NJ.

Induction of Fine-grained Part-of-speech Taggers via Classifier Combination and Crosslingual Projection

Elliott Franco Drábek
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
edrabek@cs.jhu.edu

David Yarowsky
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
yarowsky@cs.jhu.edu

Abstract

This paper presents an original approach to part-of-speech tagging of fine-grained features (such as case, aspect, and adjective person/number) in languages such as English where these properties are generally not morphologically marked.

The goals of such rich lexical tagging in English are to provide additional features for word alignment models in bilingual corpora (for statistical machine translation), and to provide an information source for part-of-speech tagger induction in new languages via tag projection across bilingual corpora.

First, we present a classifier-combination approach to tagging English bitext with very fine-grained part-of-speech tags necessary for annotating morphologically richer languages such as Czech and French, combining the extracted features of three major English parsers, and achieve fine-grained-tag-level syntactic analysis accuracy higher than any individual parser.

Second, we present experimental results for the cross-language projection of part-of-speech taggers in Czech and French via word-aligned bitext, achieving successful fine-grained part-of-speech tagging of these languages without any Czech or French training data of any kind.

1 Introduction

Most prior research in part-of-speech (POS) tagging has focused on supervised learning over a tagset such as the Penn Treebank tagset for English, which is restricted to features that are morphologically distinguished in the focus language. Thus the only verb person/number distinction made in the Brown Corpus/Penn Treebank tagset is VBZ (3rd-person-singular-present), with no corresponding person/number distinction in other tenses. Similarly, adjectives in English POS tagsets typically have no distinctions for person, number or case because such properties have no morphological surface distinction, although they do for many other languages.

This essential limitation of the Brown/Penn POS subtag inventory to morphologically realized distinctions in English dramatically simplifies the problem by reducing the tag entropy per surface form (the adjective *tall* has only one POS tag (JJ) rather than numerous singular, plural, nominative, accusative, etc. variants), increasing both the standalone effectiveness of lexical prior models and word-suffix models for part-of-speech tagging.

However, for many multilingual applications, including feature-based word alignment in bilingual corpora and machine translation into morphologically richer languages, it is helpful to extract finer-grained lexical analyses on the English side that more closely parallel the morphologically realized tagset of the second (source or target) language.

In particular, prior work on translingual part-of-speech tagger projection via parallel bilingual corpora (e.g. Yarowsky et al., 2001) has been limited to inducing part-of-speech taggers in second languages (such as French or Czech) that only assign tags at the granularity of their source language (i.e.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 49–56,
Ann Arbor, June 2005.

the Penn Treebank-granularity distinctions from English). The much richer English tagsets achieved here can allow these tagger projection techniques to transfer richer tag distinctions (such as case and verb person/number) that are important to the full analysis of these languages, using only bilingual corpora with the morphologically impoverished English.

For quickly retargetable machine translation, the primary focus of effort is overcoming the extreme scarcity of resources for the low density source language. Sparsity of conditioning events for a translation model can be greatly reduced by the availability of automatic source-language analysis. In this research we attempt to induce models for the automatic analysis of morphological features such as case, tense, number, and polarity in both the source and target languages with this end in mind.

2 Prior Work

2.1 Fine-grained part-of-speech tagging

Most prior work in fine-grained part-of-speech tagging has been limited to languages such as Czech (e.g. Hajič and Hladká, 1998) or French (e.g. Foster etc.) where finer-grained tagset distinctions are morphologically marked and hence natural for the language. In support of supervised tagger learning of these languages, fine-trained tagset inventories have been developed by the teams above at Charles University (Czech) and Université de Montréal (French). The tagset developed by Hajič forms the basis of the distinctions used in this paper.

The other major approach to fine-grained tagging involves using tree-based tags that capture grammatical structure. Bangalore and Joshi (1999) have utilized "supertags" based on tree-structures of various complexity in the tree-adjoining grammar model. Using such tags, Brants (2000) has achieved the automated tagging of a syntactic-structure-based set of grammatical function tags including phrase-chunk and syntactic-role modifiers trained in supervised mode from a treebank of German.

2.2 Classifier combination for part-of-speech tagging

There has been broad work in classifier combination at the tag-level for supervised POS tagging models. For example, Màrquez and Rodríguez (1998) have performed voting over an ensemble of decision tree and HMM-based taggers for supervised English tagging. Murata et al. (2001) have combined neural networks, support vector machines, decision lists and transformation-based-learning approaches for Thai part-of-speech tagging. In each of these cases, annotated corpora containing the full tagset granularity are required for supervision.

Henderson and Brill (1999) have approached parsing through classifier combination, using bagging and boosting for the performance-weighted voting over the parse-trees from three anonymous statistical phrase-structure-based parsers. However, as their switching and voting models assumed equivalent phrase-structure conventions for merger compatibility, it is not clear how a dependency parsing model or other divergent syntactic models could be integrated into this framework. In contrast, the approach presented below can readily combine syntactic analyses from highly diverse parse structure models by first projecting out all syntactic analyses onto a common fine-grained lexical tag inventory.

2.3 Projection-based Bootstrapping

Yarowsky et al. (2001) performed early work in the cross-lingual projection of part-of-speech tag annotations from English to French and Czech, by way of word-aligned parallel bilingual corpora. They also used noise-robust supervised training techniques to train stand-alone French and Czech POS taggers based on these projected tags. Their projected tagsets, however, were limited to those distinctions captured in the English Penn treebank inventory, and hence failed to make many of the finer grained distinctions traditionally assumed for French and Czech POS tagging, such as verb person, number, and polarity and noun/adjective case.

Probst (2003) pursued a similar methodology for the purposes of tag projection, using a somewhat expanded tagset inventory (e.g. including adjective number but not case), and focusing on target-language monolingual modeling using morpheme analysis. Cucerzan and Yarowsky (2003) addressed the problem of grammatical gender projection via the use of small seed sets based on natural gender. Another distinct body of work addresses the problem of parser bootstrapping based on syntactic dependency projection (e.g. Hwa et al. 2002), often using approaches based in synchronous parsing (e.g. Smith and Smith, 2004).

Word	Core POS	Prsn	Num.	Case	Tns/ Asp.	Pol.	Voi.
The	DT	3	PL.	NOM.			
books	NN	3	PL.	NOM.			
were	VB	3	PL.		PAST	+	ACT.
provoking	VB	3	PL.		PAST-PROG.	+	ACT.
laughter	NN	3	S.	ACC.			
with	IN						
their	DT	3	PL.	'WITH'			
curious	JJ	3	PL.	'WITH'			
titles	NN	3	PL.	'WITH'			

Figure 1: Example of fine-grained English POS tags

Word	Core POS	Prsn	Num.	Case	Tns/ Asp.	Pol.	Voice
Les	DT	3	PL.	NOM.			
livres	NN	3	PL.	NOM.			
provoquaient	VB	3	PL.		PAST-PROGR.	+	ACT.
des	DT	3	PL.	ACC.			
rires	NN	3	PL.	ACC.			
avec	IN						
ses	DT	3	PL.	'WITH'			
titres	NN	3	PL.	'WITH'			
curieux	JJ	3	PL.	'WITH'			

Figure 2: Example of fined-grained POS tags projected onto a French translation

	VB	JJ	NN	Range
Person	•	•	•	1 / 2 / 3
Number	•	•	•	SINGULAR PLURAL
Case		•	•	NOMINATIVE ACCUSATIVE GENITIVE PREPOSITION-'IN' PREPOSITION-'OF' ...
Degree		•		POSITIVE COMPARATIVE SUPERLATIVE
Tense	•			PAST PRESENT FUTURE
Perfectivity	•			+ / –
Progressivity	•			+ / –
Polarity	•			+ / –
Voice	•			ACTIVE / PASSIVE

Table 1: The fine-grained POS inventory used for English

3 Tagsets

We use Penn treebank-style part-of-speech tags as a substrate for further enrichment (for all of the experiments described here, text was first tagged using the fnTBL part-of-speech tagger (Ngai and Florian, 2001)). Each Penn tag is mapped to a core part-of-speech tag, which determines the set of fine-grained tags further applicable to each word. The fine-grained tags applicable to nouns, verbs, and adjective are shown in Table 1. This paper concentrates on these most important core parts-of-speech.

The example English sentence in Figure 1 illustrates several key points about our tagset. Some of the information we are interested in is already expressed by the Penn-style tags – the NN *titles* is plural; the VBD *were* is in the past tense. For these, our goal is simply to make these facts explicit.

On the other hand, *curious* could also be meaningfully said to be semantically plural, and most importantly for us, the corresponding word in a translation of this sentence into many other languages would be morphologically plural. Similarly, the head verb *provoking* is also semantically in the past tense, and is likely to be translated to a past-tense form in many languages, even though in this example the actual tense marking is on *were*. We expect the 'pastness' of the action to be much more stable cross-linguistically, than the particular division of labor between the head word and the auxiliary. By propagating these features from where they are explicit to where they are not, we hope to make information more directly available for projection. Another important class of information we would like to make available concerns syntactic relations, which many languages mark with morphological case. This is an issue that involves deep, complex, and ambiguous mappings, which we are not yet prepared to treat in their fullness. For now, we observe that *curious* and *titles* are both dominated by *with*.

Because of intent to mark whatever information is recoverable, some of our tags require some interpretation. For example, English has little or no morphological realization of syntactic case, but the essential information of case, relationship of a noun with its governor, is recoverable from contextual information, so we defined it in these terms. To avoid loss of information, we chose to remain agnostic about deeper analyses, such as the identification of theta roles or predicate-argument relationships, and restricted ourselves to a direct representation of surface relationships. We identified subjects, direct and indirect objects, non-heads of noun compounds, possessives, and temporal adjuncts, and created a distinct tag for the objects of each distinct preposition.

Our ideal would be to have as expansive and detailed a tagset as possible, a 'quasi-universal' tagset which could cover whatever set of distinctions might be relevant for any language onto which we might

Feature	Antecedent → CONSEQUENT
Noun Number	NN → SINGULAR NNS → PLURAL
Verb Tense	VBD → PAST (will|shall) RB* VB → FUTURE

Figure 3: Examples of locally recoverable features

project our analysis. A completely universal tagset would require that the morphological distinctions made by the world's languages come from a limited pool of possibilities, based on non-arbitrary semantic distinctions, and further would require that the relevant semantic information be recoverable from English text. The tagset we are using now is shaped in part by exceptions to these conditions. For example, we have put off implementing tagging of gender given the notoriously arbitrary and inconsistent assignment of grammatical gender across languages (although Cucerzan and Yarowsky (2003) were able to show success on projection-based analysis of grammatical gender as well).

In the end, we have settled on a set of distinctions very similar to those realized by the morphologically richer of the European languages, with the noticeable absence of gender. Table 1 describes the features we chose on this basis (definiteness and mood features were developed for English but not projected to French or Czech, and are not treated in this paper).

4 Methods – English Tagging

The features we tagged vary widely in their degree of morphological versus syntactic marking, and the difficulty of their monolingual English detection. For some, tagging is simply a matter of explicitly separating information contained in the Penn part-of-speech tags, while others can be tagged to a high degree of accuracy with simple heuristics based on local word and part-of-speech tag patterns. These include number for nouns and adjectives, person (trivially) for nouns, degree for adjectives, polarity, voice, and aspect (perfectivity and progressivity) for verbs, as well as tense for some verbs. Figure 3 shows example rules for some of these easier cases.

The more difficult features are those whose detection requires some degree of syntactic analysis. These include case, which summarizes the relation of each noun with its governor, and the agreement-based features: we define person, number, and case for attributive adjectives by agreement with their head nouns, number and person for verbs and predicate adjectives by agreement with their subjects, and tense for some verbs by agreement with their inflected auxiliaries.

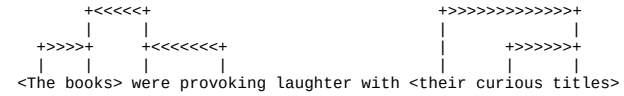
We investigated four individual approaches for the syntax-features – a regular-expression-based quasi-parser, a system based on Dekang Lin's MiniPar (Lin, 1993), a system based on the Collins parser (Collins, 1999), and one based on the CMU Link Grammar Parser (Sleator and Temperley, 1993), as well as a family of voting-based combination schemes.

4.1 Regular-expression Quasi-parser

The regular-expression 'quasi-parser' takes a direct approach, using several dozen heuristics based on regular-expression-like patterns over words, Penn part-of-speech tags, and the output of the fnTBL noun chunker. Use of the noun chunker facilitates identification of noun/dependent relationships within chunks, and extends the range of patterns identifying noun/governor relationships across chunks.

The output of the quasi-parser consists of two parts: a case tag for each noun in a sentence, and a set of agreement links across which other features are then spread. We call this a direct approach because the links are defined operationally, directly indicating the spreading action, rather than representing any deeper syntactic analysis.

In the diagram of the example sentence below, an arrow from one word to another indicates that the former takes features from the latter. The example also shows the context patterns by which the nouns in the sentence receive case.



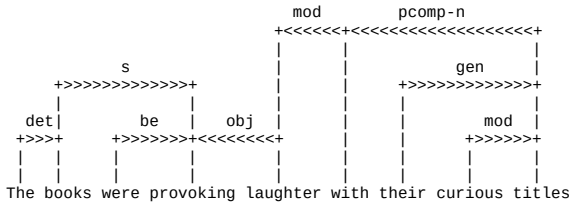
Word	Context Pattern → CASE TAG
laughter	VB (genitive-NP)* • → ACCUSATIVE
titles	with (genitive-NP)* • → PREP-WITH
books	*default* → NOMINATIVE

4.2 MiniPar and the CMU Link Grammar Parser

For MiniPar, the Collins parser, and the CMU link grammar parser, we developed for each a set of minimal-complexity heuristics to transform the parser output into the specific conceptions of dependency and case we had developed for the first pass.

MiniPar produces a labeled dependency graph, which yields a straightforward extraction of the information needed for this task. Case tagging is a simple matter of mapping the set of dependency labels to our case inventory. Our agreement links are almost a subset of MiniPar's dependencies (with some special treatment of subject/auxiliary/main-verb triads, as shown in the example sentence).

The figure below presents MiniPar's raw output for the example sentence, along with some example dependency-label/case-tag rules. The agreement links extracted from the dependency graph are identical (in this case) to those produced by the regular-expression quasi-parser.



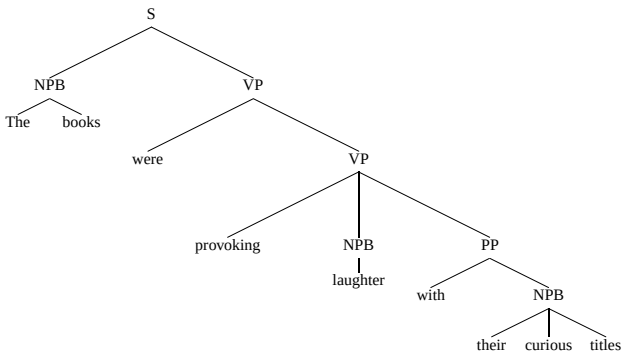
Word	`Dependency Label` → CASE TAG
books	`s` → NOMINATIVE
laughter	`obj` → ACCUSATIVE
titles	`pcomp-n:with` → PREP-WITH

The output of the CMU link grammar parser has properties similar to MiniPar, and thus tag extraction was handled in a similar fashion.

4.3 Collins Parser

The Collins Parser produces a Penn-Treebank-style constituency tree, with head labels. Although we could have used the head-labels to operate on the dependency graph as with MiniPar, we chose to concentrate on addressing the weakest point of our previous systems, the identification of case. Our algorithm traces the path from each noun to the root of the tree, stopping at the first node which we judged to reliably indicate case.

We did not directly extract any further information from the Collins parser output. Instead, the remainder of the system is identical to the regular-expression quasi-parser. However, because the system uses nominative case to identify verb subjects, we did expect to see some improvements in agreement-based features as well.



Word	`Path to Root` → CASE TAG
books	`NPB:S` → NOMINATIVE
laughter	`NPB:VP:VP:VP:S` → ACCUSATIVE
titles	`NPB:PP(with):VP:VP:S` → PREP-WITH

4.4 Parser Combination

The fine-grained taggers based on the four participating parsers exhibited significant differences in their strengths and weaknesses, suggesting potential benefit from combining them. Lacking tag-level numerical scores and development data for weight-training, we restricted ourselves to simple voting mechanisms. We chose to do all of the combinations at the end of the process, voting separately on tags for specific features of specific words. Without tag-level probabilities from the one-best parser outputs, we were still able to use the combination protocols to achieve a coarse-grained confidence measure.

We compared a series of seven combination protocols of increasing leniency to investigate precision/recall tradeoffs. The strictest, '4:0', produces an output only when there are four votes for the favored tag, and no votes for any other. Analogously, protocols '3:0', '2:0' and '1:0' also allow no dissent, but allow progressively more abstentions. Continuing the sequence, protocol '2:1' proposes a tag as long as there is a clear majority, '2:2' as long as supporters are not outnumbered by dissenters, and '1:3' whenever possible. To break ties in the latter two protocols, we favored first the CMU Link Parser, then Collins, then MiniPar, then Regexp. (Lacking sufficient labeled data for fine-tuning, we ordered them arbitrarily.)

5 Evaluation of English POS Tagging

Before we began the development of our taggers, we created standard tagging guidelines, and hand annotated a 3013-word segment of the English side of the Canadian Hansards, to be used for evaluation.

Core POS	Feature	MiniPar	Regexp	Collins	CMU Link	1:3
JJ	num	86.8	87.7	87.7	87.9	**88.4**
	case	65.1	74.5	76.4	79.2	**80.6**
	deg	**100**	**100**	**100**	**100**	**100**
	'French'	86.8	87.7	87.7	87.9	**88.4**
	'Czech'	57.9	64.3	67.1	68.1	**70.5**
NN	num	**99.7**	**99.7**	**99.7**	**99.7**	**99.7**
	case	65.9	74.8	77.8	77.3	**80.0**
	'French'	**99.7**	**99.7**	**99.7**	**99.7**	**99.7**
	'Czech'	65.0	74.8	77.8	77.2	**79.9**
VB	num	77.2	64.8	65.5	66.8	**78.1**
	tns	**77.2**	66.8	67.1	67.1	76.3
	prsn	**88.0**	75.0	74.3	73.4	86.5
	pol	96.3	**96.6**	**96.6**	**96.6**	**96.6**
	voice	**88.0**	**88.0**	**88.0**	**88.0**	**88.0**
	'French'	61.8	61.3	61.0	61.3	**67.5**
	'Czech'	61.3	61.1	60.8	61.1	**67.1**
overall	*'French'*	82.6	82.5	82.4	83.2	**85.2**
	'Czech'	62.5	67.8	69.4	70.5	**73.3**

Table 2: English tagging forced-choice accuracy

Core POS	Feature	Mini Par	Regexp	Collins	CMU Link	2:0	1:0	1:2
JJ	num	79.1	81.3	81.3	82.2	81.2	83.8	**83.9**
	case	72.1	79.2	83.0	78.9	78.1	79.1	**84.2**
	deg	100	100	100	100	100	100	**100**
	'Czech'	67.6	72.2	76.0	74.3	70.4	73.4	**77.9**
NN	num	99.7	99.7	99.7	99.7	99.7	99.7	**99.7**
	case	68.5	75.5	**78.6**	77.9	72.6	72.5	78.1
	'Czech'	68.1	75.2	**78.3**	77.7	72.2	72.1	77.8
VB	tns	78.0	68.5	68.7	68.0	68.7	78.3	**78.3**
	num	72.7	61.3	61.2	61.3	61.1	76.1	**77.1**
	prsn	77.2	66.5	65.4	63.9	64.0	78.3	**79.0**
	pol	96.3	96.6	96.6	96.5	96.5	96.5	**96.6**
	voice	88.0	88.0	88.0	88.0	88.0	88.0	**88.0**
	'French'	61.7	50.7	50.2	50.1	50.6	64.8	**65.6**
	'Czech'	61.1	50.5	49.9	49.8	50.4	64.5	**65.2**
all	*'French'*	81.9	78.7	78.5	78.5	83.6	78.9	**83.9**
	'Czech'	65.4	66.0	67.8	69.3	68.9	63.5	**72.9**

Table 3: English tagging F-measure

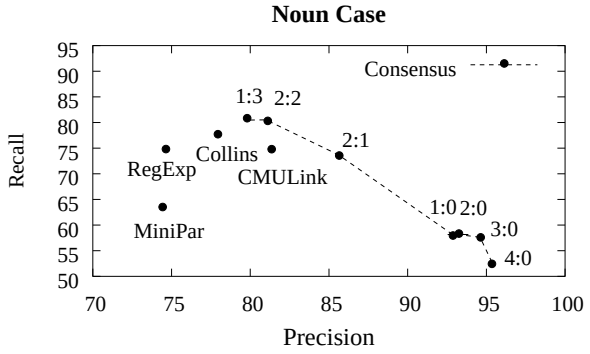


Figure 4: Precision versus Recall – Noun case

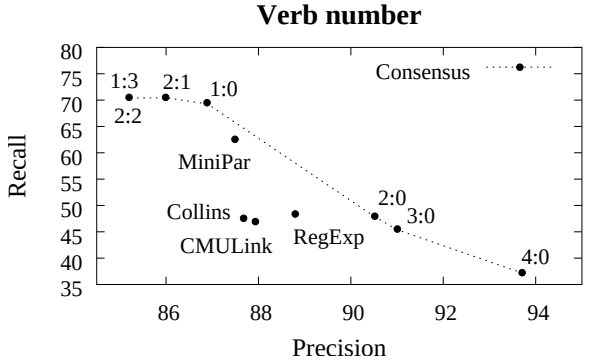


Figure 5: Precision versus Recall – Verb number

Table 2 shows system accuracy on test data in a forced-choice evaluation, where abstentions were replaced by the most common tag for the each situation (the combination system is that one biased most heavily towards recall.)

In addition to the individual features, we also list 'pseudo-French' and 'pseudo-Czech'. These represent exact-match accuracies for composite features comprising those features typically realized in French or Czech POS taggers. For example, pseudo-Czech verb accuracy of 67.1% indicates that for 67.1% of verb instances, the Czech-realized features of number, tense, perfectivity, progressivity, polarity, and voice were *all* correct. These give an indication of the quality of the starting point for crosslingual bootstrapping to the respective languages.

Besides the forced-choice scenario, we were also interested in the effect of allowing abstentions for low-confidence cases. Table 3 shows the F-measure of precision and recall for the individual systems, as well as a range of combination systems. Figures 4 and 5 show (for two example features) the clear precision/recall tradeoff. Performance of the consensus systems is higher than the individual parser-based taggers at all levels of tag precision or recall.

Unfortunately, because MiniPar does its own integrated tokenization and part-of-speech tagging, we found that a significant portion of the errors seemed to stem from discrepancies where MiniPar disagreed on the segmentation or the core part-of-speech of the words in question.

6 Cross-lingual POS Tag Projection and Bootstrapping

Our cross-lingual POS tag projection process is similar to Yarowsky et al. (2001). It begins by performing a statistical sentence and word alignment of the bilingual corpora (described below), and then transfers both the coarse- and fine-grained tags achieved from classifier combination on the English side via the higher confidence word alignments (based on the intersection of the 1-best word alignments induced from French to English and English to French. The projected tags then serve as noisy monolingual training data in the source language.

There are several notable differences and extensions: The first major difference is that the projected fine-grained tag set is much more detailed, including such additional properties as noun case, adjective

case and number, and verb person, number, voice, and polarity. Because these span the subtag features normally assumed for Czech and French part-of-speech taggers, the projection work presented here for the first time shows the translingual projection and induction of full-granularity Czech and French taggers, rather than the much less complete and coarser-grained prior projection work.

The other major differences are in the method of target-language monolingual tagger generalization from the projected tags. We pursue a combination of trie-based lexical prior models and local-agreement-based context models. The lexical prior trie model, as illustrated in Figure 6 for noun number, shows how the hierarchically-smoothed lexical prior conditioned on variable length suffixes can assign noun number probabilities to both previously seen words (with full-word-length suffixes stored) and to new words in test data, based on backoff to partially matching suffixes.

The context models are based on exploiting agreement phenomena of the fine-grained tag features in local context. $P(subtag|context)$ for each word token is a distance-weighted linear interpolation of the posterior tag distributions assigned to its neighbors by the trie-based lexical-prior model. Finally $P(\text{subtag}|\text{word})$ is an equally-weighted linear interpolation of the $P(\text{subtag}|\text{affix})$ trie model probability and $P(\text{subtag}|\text{context})$ context-agreement probability. Table 4 contrasts the performance of these two models in isolation and combination.

All of these models condition their probabilities first on the core part-of-speech of a word. We used the methods of Yarowsky et al. (2001) to develop a core part-of-speech tagger for French, based only on the projected core tags, and used this as a basis for fine-grained tags. We also ran experiments isolating the question of fine-grained tagging, assuming as input externally supplied core tags from the gold-standard data. Table 4 shows results under both of these assumptions.

For French, the training data was 15 million words from the Canadian Hansards. Word alignments were produced using GIZA++ (Och and Ney, 2000) set to produce a maximum of one English word link for each French word (i.e., a French-to-English model). The test data was 111,000 words of text from the Laboratoire de Recherche Appliquée en Linguistique Informatique at the Université de Montréal, annotated with person, number, and tense.

Suffix	**Pr(**PLURAL|**suffix)**	**Pr(**SINGULAR|**suffix)**
none	32.5	67.5
-s	66.5	33.5
-is	35.3	64.7
-ais	16.2	83.8

Figure 6: Example smoothed suffix trie probabilities for French noun number

Several factors contributed to a fairly successful set of results. The quality of the alignments is subjectively very good; the morphological system of French is relatively simple, and is a good match for our suffix tries; Perhaps most importantly, the mappings between the English and the French tagsets were for the most part simple and consistent. The most prominent exception is verb tense.

For Czech, the training and testing data were from the Reader's Digest corpus. We used the first 63,000 words for testing, and the remaining 551,000 for training, ignoring the translations of the test data and the gold-standard tags on the training data.

It should be noted that the *baseline* (most likely tag) performance is actually a supervised model using the target language monolingual goldstandard data frequencies. The other results based on translingual projection have no knowledge of the true most likely tag, and hence occasionally underperform this supervised "baseline". Finally, one of the major reasons for lower Czech performance is the currently very poor quality of the bilingual word alignments. However, using these diverse POS subtags as features offers the potential for substantially improved word alignment for morphologically rich languages, one of the central downstream benefits of this research.

7 Conclusion

We have demonstrated the feasibility of automatically annotating English text with morphosyntactic information at a much finer POS tag granularity than in the standard Brown/Penn tagset, but at a POS detail appropriate for tagging morphologically richer language such as Czech or French. This is accomplished by using a classifier combination strategy to integrate the analyses of four independent parsers, achieving a consensus tagging with higher accuracy than the best component parser.

Furthermore, we have demonstrated that the resulting fine-grained POS tags can be successfully

Feature	Engl. Comb.	Baseline	Trie	Vic.	Comb.
French (using correct core POS)					
JJ-num	1:0	67.0	97.6	98.0	**98.2**
	2:0	67.0	97.6	98.0	**98.2**
NN-num	1:0	71.2	94.3	**94.7**	94.6
	2:0	71.2	94.3	**94.7**	94.6
VB-num	1:0	53.4	**91.9**	73.2	90.2
	2:0	53.4	73.1	72.7	**73.2**
VB-prsn	1:0	**88.0**	76.9	78.7	77.7
	2:0	88.0	92.9	93.0	**93.4**
VB-tns	1:0	47.6	**86.2**	71.7	73.9
	2:0	47.6	**54.7**	51.9	53.8
VB-exact	1:0	26.8	**48.1**	43.4	47.1
	2:0	26.8	**50.0**	46.9	49.2
overall-exact	1:0	56.2	**79.7**	78.5	79.6
	2:0	56.2	80.3	79.6	**80.3**
French (induced core POS)					
JJ-num	1:0	65.1	87.1	**89.0**	88.3
	2:0	65.1	87.1	**89.1**	88.5
NN-num	1:0	66.6	87.5	87.8	**87.9**
	2:0	66.6	87.5	87.8	**87.9**
VB-num	1:0	53.0	**86.4**	79.5	84.9
	2:0	53.0	71.2	70.6	**71.4**
VB-prsn	1:0	**75.1**	67.4	69.7	68.4
	2:0	75.1	80.4	80.8	**81.1**
VB-tns	1:0	43.3	**65.1**	62.0	64.2
	2:0	43.3	**49.0**	46.3	48.2
VB-exact	1:0	24.1	**43.9**	40.2	43.0
	2:0	24.1	**45.3**	42.2	44.6
overall-exact	1:0	52.6	73.3	72.5	**73.4**
	2:0	52.6	73.7	73.1	**73.9**
Czech (using correct core POS)					
JJ-num	1:0	28.0	**46.4**	44.5	45.1
	2:0	28.0	**47.0**	44.6	46.0
JJ-case	1:0	7.1	40.2	**42.0**	40.9
	2:0	7.1	37.9	**41.4**	40.2
JJ-deg	1:0	89.2	85.6	**86.8**	86.6
	2:0	89.2	85.6	**86.8**	86.6
JJ-exact	1:0	6.9	**20.6**	19.1	19.4
	2:0	6.9	**20.9**	20.0	20.5
NN-num	1:0	52.2	**71.1**	69.6	70.7
	2:0	52.2	**71.1**	69.4	70.8
NN-case	1:0	**53.5**	39.5	39.2	39.6
	2:0	**53.5**	39.2	38.6	39.1
NN-exact	1:0	23.7	**29.5**	28.7	29.4
	2:0	23.7	**29.7**	28.6	29.4
VB-num	1:0	57.0	**71.6**	69.1	70.7
	2:0	57.0	71.2	69.7	**71.4**
VB-prsn	1:0	55.1	**65.9**	64.9	65.4
	2:0	55.1	**65.3**	64.3	64.9
VB-voice	1:0	**97.3**	93.2	93.9	93.4
	2:0	**97.3**	93.2	93.9	93.4
VB-pol	1:0	91.1	**93.8**	89.9	92.1
	2:0	91.1	**93.8**	89.9	92.1
VB-exact	1:0	9.9	**15.2**	14.6	14.8
	2:0	9.9	14.5	14.3	**14.7**
overall-exact	1:0	15.7	**22.6**	21.8	22.2
	2:0	15.7	**22.5**	21.7	22.3

Table 4: **Accuracy of induced fine-grained taggers**, by core part-of-speech, feature, underlying english tagger combination (*eng-comb.*), and french tagging method (most likely tag – *baseline*, suffix trie (prefix trie for Czech verb polarity) – *trie*, vicinity voting – *vic.*, or trie/vicinity combination – *comb.*)

projected to additional languages such as French and Czech, generating stand-alone taggers capturing the salient fine-grained POS subtag distinctions appropriate for these languages, including features such as adjective number and case that are not morphologically marked in the original English.

References

S. Bangalore and A. K. Joshi. 1999. Supertagging: An Approach to Almost Parsing. *Computational Linguistics*, 25(2): 237–265.

T. Brants. 2000. TnT – a Statistical Part-of-Speech Tagger. In *Proceedings of ANLP-2000*, pp. 224–231.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. Dissertation, University of Pennsylvania, 1999.

S. Cucerzan and D. Yarowsky. 2003. *Minimally Supervised Induction of Grammatical Gender*. In *Proceedings of HLT/NAACL-2003*, pp. 40–47.

J. Hajič and B. Hladk´a. 1998. Tagging inflective languages: prediction of morphological categories for a rich, structured tagset. In *Proceedings of COLING-ACL Conference*, pp. 483–490.

R. Hwa, P. Resnik, and A. Weinberg. 2002. Breaking the Resource Bottleneck for Multilingual Parsing. In *Proceedings of LREC-2002*.

D. Lin. 1993. Principle-based parsing without overgeneration. In *Proceedings of ACL-93*, pp. 112–120.

L. Màrquez and H. Rodr´ıguez. 1998. Part-of-speech tagging using decision trees. In *Proceedings of the European Conference on Machine Learning*.

M. Murata, Q. Ma, and H. Isahara. 2001. Part of Speech Tagging in Thai Language Using Support Vector Machine. In *Proceedings of NLPNN-2001*, pp. 24–30.

G. Ngai and R. Florian. 2001. Transformation-based Learning in the Fast Lane. In *Proceedings of NAACL-2001*, pp. 40–47.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL-2000*, pp. 440–447.

K. Probst. 2003. Using 'smart' bilingual projection to feature-tag a monolingual dictionary. In *Proceedings of CoNLL-2003*, pp. 103–110.

D. Sleator and D. Temperley. 1993. Parsing English with a Link Grammar. In *Proceedings, Third International Workshop on Parsing Technologies*, pp. 277–292.

D. Smith and N. Smith 2004. Bilingual Parsing with Factored Estimation: Using English to Parse Korean. In *Proceedings of EMNLP-2004*, pp. 49–56.

D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT-2001, First International Conference on Human Language Technology Research*, pp. 161–168.

A hybrid approach to align sentences and words in English-Hindi parallel corpora

Niraj Aswani
Department of Computer Science
University of Sheffield
Regent Court 211, Portobello Street
Sheffield S1 4DP, UK
N.Aswani@dcs.shef.ac.uk

Robert Gaizauskas
Department of Computer Science
University of Sheffield
Regent Court 211, Portobello Street
Sheffield S1 4DP, UK
R.Gaizauskas@dcs.shef.ac.uk

Abstract

In this paper we describe an alignment system that aligns English-Hindi texts at the sentence and word level in parallel corpora. We describe a simple sentence length approach to sentence alignment and a hybrid, multi-feature approach to perform word alignment. We use regression techniques in order to learn parameters which characterise the relationship between the lengths of two sentences in parallel text. We use a multi-feature approach with dictionary lookup as a primary technique and other methods such as local word grouping, transliteration similarity (edit-distance) and a nearest aligned neighbours approach to deal with many-to-many word alignment. Our experiments are based on the EMILLE (Enabling Minority Language Engineering) corpus. We obtained 99.09% accuracy for many-to-many sentence alignment and 77% precision and 67.79% recall for many-to-many word alignment.

1 Introduction

Text alignment is not only used for the tasks such as bilingual lexicography or machine translation but also in other language processing applications such as multilingual information retrieval and word sense disambiguation. Whilst resources like bilingual dictionaries and parallel grammars help to improve Machine Translation (MT) quality, text alignment, by aligning two texts at various levels (i.e. documents, sections, paragraphs, sentences and words), helps in the creation of such lexical resources (Manning & Schütze, 2003).

In this paper, we describe a system that aligns English-Hindi texts at the sentence and word level. Our system is motivated by the desire to develop for the research community an alignment system for the English and Hindi languages. Building on this, alignment results can be used in the creation of other Hindi language processing resources (e.g. part-of-speech taggers). We present a simple sentence length approach to align English-Hindi sentences and a hybrid approach with local word grouping and dictionary lookup as the primary techniques to align words.

2 Sentence Alignment

Sentence alignment techniques vary from simple character-length or word-length techniques to more sophisticated techniques which involve lexical constraints and correlations or even cognates (Wu 2000). Examples of such alignment techniques are Brown et al. (1991), Kay and Roscheisen (1993), Warwick et al. (1989), and the "align" programme by Gale and Church (1993).

2.1 Length-based methods

Length-based approaches are computationally better, while lexical methods are more resource

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 57–64,
Ann Arbor, June 2005.

hungry. Brown et al. (1991) and Gale and Church (1993) are amongst the most cited works in text alignment work. Purely length-based techniques have no concern with word identity or meaning and as such are considered knowledge-poor approaches. The method used by Brown et al. (1991) measures sentence length in number of words. Their approach is based on matching sentences with the nearest length. Gale and Church (1993) used a similar algorithm, but measured sentence length in number of characters. Their method performed well on the Union Bank of Switzerland (UBS) corpus giving a 2% error rate for 1:1 alignment.

2.2 Lexical methods

Moving towards knowledge-rich methods, lexical information can be vital in cases where a string with the same length appears in two languages. Kay and Roscheisen (1993) tried lexical methods for sentence alignment. In their algorithm, they consider the most reliable pair of source and target sentences, i.e. those that contain many possible lexical correspondences. They achieved 96% coverage on Scientific American articles after four passes of the algorithm. Other examples of lexical methods are Warwick et al. (1989), Mayers et al. (1998), Chen (1993) and Haruno and Yamazaki (1996).

Warwick et al. (1989) calculate the probability of word pairings on the basis of frequency of source word and the number of possible translations appearing in target segments. They suggest using a bilingual dictionary to build word-pairs. Mayers et al. (1998) propose a method that is based on a machine readable dictionary. Since bilingual dictionaries contain base forms, they pre-process the text to find the base form for each word. They tried this method in an English-Japanese alignment system and got accuracy of about 89.5% for 1-to-1 and 42.9% for 2-to-1 sentence alignments. Chen (1993) constructs a simple word-to-word translation model and then takes the alignment that maximizes the likelihood of generating the corpus given the translation model. Haruno and Yamazaki (1996) use a POS tagger for source and target languages and use an online dictionary to find matching word pairs. Haruno and Yamazaki (1996) pointed out that though dictionaries cannot capture context dependent keywords in the corpus, they can be very useful to obtain information about words that appear only once in the corpus. Lexical methods for sentence alignment may also result in partial word alignment. Given that lexical methods can be computationally expensive, our idea was to try a simple length-based approach similar to that of Brown et al. (1991) for sentence alignment and then use lexical methods to align words within aligned sentences.

2.3 Algorithm

We use English-Hindi parallel data from the EMILLE corpus for our experiments. EMILLE is a 63 Million word electronic corpus of South Asian languages, especially those spoken as minority languages in UK. It has around 120,000 words of parallel data in each of English, Hindi, Urdu, Punjabi, Bengali, Gujarati, Sinhala and Tamil (Baker et al., 2004).

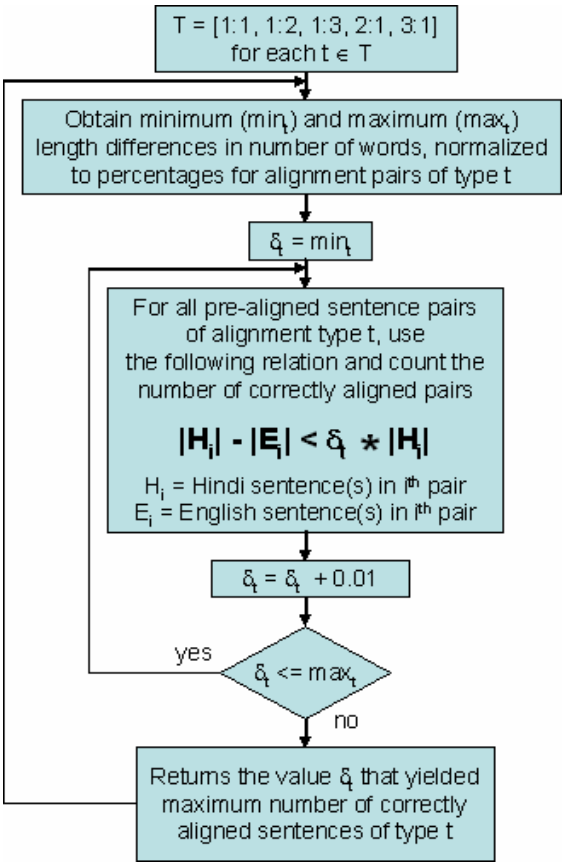


Figure 2.1 Sentence Alignment Parameter Learning algorithm

Table 2.1 Rules for the Sentence Alignment Algorithm

Rule	If	Hindi:English Alignment
H1	$\|h_i\| - (\|e_j\| + \|e_{j+1}\|) < 0.17 * \|h_i\|$	1-To-2
H2	$\|h_i\| - (\|e_j\| + \|e_{j+1}\| + \|e_{j+2}\|) < 0.17 * \|h_i\|$	1-To-3
E1	$\|e_j\| - (\|h_i\| + \|h_{i+1}\|) < 0.17 * \|e_j\|$	2-To-1
E2	$\|e_j\| - (\|h_i\| + \|h_{i+1}\| + \|h_{i+2}\|) < 0.14 * \|e_j\|$	3-To-1
Default	$(\|e_j\| = \|h_i\|)$ || (Rule H1 and E1 Fails)	1-To-1

Examining the data, we observe that it is possible to align one English sentence with one or more Hindi sentences or vice-versa. In the method described below, sentence *length* is calculated in *number of words*. We define our task as that of learning rules that characterise the relationship between the lengths of two sentences in parallel texts. We used 60 manually aligned paragraphs from the EMILLE corpus, each with an average of 3 sentences, as a dataset for our learning task. Initially we derived minimum and maximum length differences in percentages for each of the one-to-one, one-to-two and one-to-three parallel sentence pairs. Later we used these values as input to our algorithm to learn new rules that maximize the probability of aligning sentences.

Learning: Let T = [1:1, 1:2, 1:3, 2:1, 3:1], a set of possible alignment types between the English and Hindi sentences. For each alignment type $t \in T$, minimum and maximum length differences in number of words, normalized to percentages, can be described as min_t and max_t. For each alignment type $t \in T$, a constant parameter δ_t, where $\delta_t \in [min_t , min_t + 0.01, min_t + 0.02, \ldots, max_t]$ was learned using an algorithm described in figure 2.1. δ_t is a value that describes the length relationship between the sentences of a pair of type *t*. For example, given a pair of one Hindi and two English sentences and a value δ_t, where t = 1:2, it is possible to check if these sentences can be aligned with each other. Suppose for a given pair of parallel sentences that consist of h_i (Hindi sentence at i^{th} position) and e_j and e_{j+1} (English sentences at j^{th} and $j+1^{th}$ positions), let $|h_i|$, $|e_j|$ and $|e_{j+1}|$ be the lengths of Hindi and English sentences. h_i, e_j and e_{j+1} are said to have 1:2 alignment if $\mathbf{|h_i| - (|e_j| + |e_{j+1}|) < 0.17 * |h_i|}$, i.e. the difference between the length of the Hindi sentence and the length of the two consecutive English sentences is less than ($\delta_{t=1:2}$ = 0.17) times the length of the Hindi sentence. Table 2.1 lists rules for different possible alignments. Before we decide on the final alignment, we check each possibility of one Hindi sentence being aligned with one, two or three consecutive English sentences and vice-versa. We use rules H1 and H2 to check the possibility of one Hindi sentence being aligned with two or three consecutive English sentences. Similarly, rules E1 and E2 are used to check the possibility of one English sentence being aligned with two or three consecutive Hindi sentences. If none of the rules from H1, H2, E1 and E2 return true, we consider the default alignment (1-To-1) between the English and Hindi sentences. We give preference to the higher alignment over the possible lower alignments, i.e. given 1-To-2 and 1-To-3 possible alignment mappings, we consider 1-To-3 mapping. We tested our algorithm on parallel texts with total of 3441 English-Hindi sentence pairs and obtained an accuracy of 99.09%; i.e., the correctly aligned pairs were 3410.

3 Word Alignment

Extending sentence alignment to word alignment is a process of locating corresponding word pairs in two languages. In some cases, a word is not translated, or is translated by several words. A word can also be a part of an expression that is translated as a whole, and therefore the entire expression must be translated as a whole (Manning & Schütze, 2003). We present a hybrid method for many-to-many word alignment. Hindi is a partial free order language where the order of word groups in a Hindi sentence is not fixed, but the order of words within groups is fixed (Ray et al., 2003). According to Ray et al. (2003), fixed order word group extraction is essential for decreasing the load on the free word order parser. The word alignment algorithm takes as input a pair of aligned sentences and groups words in sentences of both languages. We have observed a few facts about the Hindi language. For example, there are no

articles in Hindi (Bal Anand, 2001). Since there are no articles in Hindi, articles are aligned to null.

3.1 Local word grouping

A separate group is created for each token in the English text. Every English word has one property associated with it: the lemma of the word. This is necessary because a dictionary lookup approach is at the heart of our word alignment algorithm. Verbs are used in different inflected forms in different sentences. For a verb, it is common not to find all inflected forms listed in a dictionary, i.e. most dictionaries contain verbs only in their base forms. Therefore we use a morphological analyzer to find the lemma of each English word.

Word groups in Hindi are created using two resources: a Hindi gazetteer list that contains a large set of named entities (NE) and a rule file that contains more than 250 rules. The gazetteer list is available as a part of Hindi Gazetteer Processing Resource in GATE (Maynard et al., 2003). For each rule in the rule file, it contains the following information:

1. Hindi Regular Expression (RE) for a word or phrase. This must match one or more words in the Hindi sentence.
2. Group name or a part-of-speech category.
3. Expected English word(s) (EEW) that this Hindi word group may align to.
4. Expected Number of English words (NW) that the Hindi group may align to.
5. In case a group of one or more English words aligns with a group of one or more Hindi words, information about the key words (KW) in both groups. Key words must match each other in order to align English-Hindi groups.
6. A rule to convert the Hindi word into its base form (BF).

Rules in the rule file identify verbs, postpositions, noun phrases and also a set of words, whose translation is expected to occur in the same order as the English words in the English sentence. The local word grouping algorithm considers one rule at a time and tries to match the regular expression in the Hindi sentence. If the expression is matched, a separate group for each found pattern is created. When a Hindi group is created, based on its pattern type, one of the following categories is assigned to that group:

proper-noun	city	job-title	location
country	number	day-unit	date-unit
month-unit	verb	auxiliary	pronoun
post-position	other		

These rules have been obtained mainly through consulting Hindi grammar material (Bal Anand, 2001 and Ta, 2002) and by observing the EMILLE corpus. For example, consider the following rules:

No	RE	Cat	EEW	NW	KW	BF
1	बावन	num	fifty two	2		
2	(.)+ रहा	verb			1	
3	(.)+ ते थे	verb			1	1,ते = ना
4	(.)+ के लिये	prep	for (.)+	2	1-2	
5	अलग अलग	other	different	1		

i) "रहा ", "रहे ,""रही" are used to indicate the progressive tense. They can be seen as analogous to the English (-ing) ending.
ii) "ते ,""ता", and "ती" are used as verb endings to indicate the habitual tense. They must agree with subject number and gender.
iii) "थे "is a past tense conjunction of the verb "होना".

In the first rule, if we find a word "बावन" (bavan) in Hindi, we mark it as a "Number" and search for the English string with two words that is equal to the expected string "fifty two". In the second rule, we locate a string where the second word is "रहा" (raha). "1" in the fifth column specifies that the first word is the keyword. We use the dictionary to locate the word in the English sentence that matches with the key word. If the English word is located, we align "(.)+ रहा" with the English word found. In the third rule, if we find a Hindi string with two words where the first word ends with "ते " (te) and the second word is "थे "(the), we group them as a verb. As specified in the sixth column, we replace the characters "ते "with "ना" (na) to convert the first word into its base form (e.g. "गाते "(gaate) into "गाना" (gaana)). In the fourth rule, we align "X के लिये" with "For X", where "For" = "के लिये". As specified in the fifth column, we align the first word in Hindi with the second word in English. In the final example, we group two words that are identical to each other. For example: "अलग अलग" (alag alag) which means "different" in English. Such bigrams are used to stress the importance of a word/activity in a sentence.

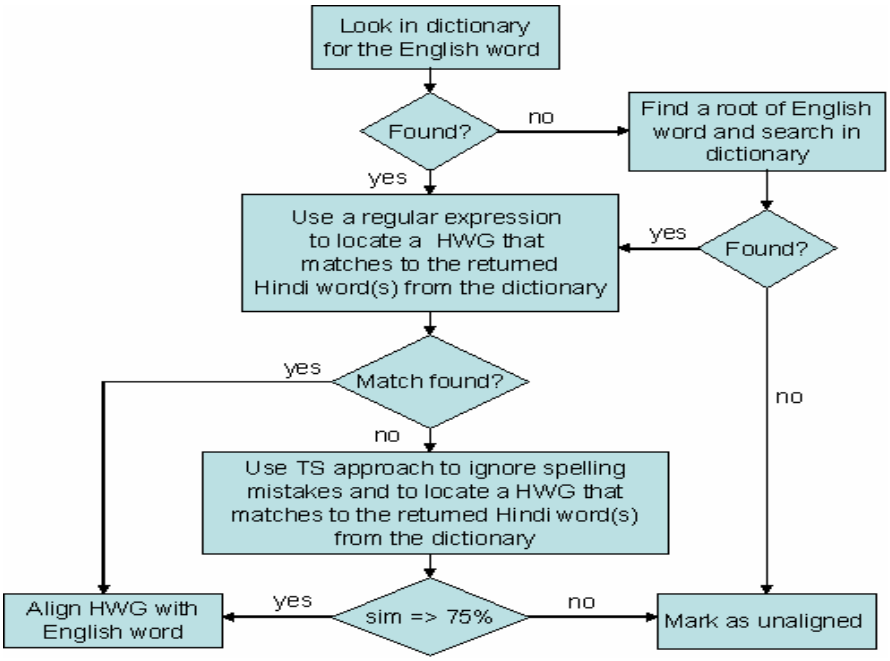


Figure 3.1 Dictionary Lookup Approach

example, in rule 3 and 4 if the word ends with either of ता, ते or ती followed by (PH), it is assumed that the word is a verb. The formula for finding the lemma of any Hindi verb is: **infinitive = root verb + "ना"**. Sometimes it is possible to predict the corresponding English translation. For example, for the postposition "के सामने", one is likely to find the preposition "in front of" in the English sentence. We store this information as an expected English word(s) in Hindi Word Groups (HWGs) and search for it in the English sentence. In the case of rules 4 and 5, though the HWG contains more than one word, only one is the actual verb (key word) that is expected to be available in a dictionary. We specify the index of this key word in the HWG, so as to consider only the word at the specified index to compare with key word in English word group. If they match, the full HWG is aligned to the word in English sentence.

3.2 Alignment Algorithm

After applying the local word grouping rules to the Hindi sentence(s), based on their categories of HWGs, we use four methods to process and align HWGs with their respective English Word Groups.

1. Dictionary lookup approach (DL)
2. Transliteration similarity approach (TS)
3. Expected English words approach (EEW)
4. Nearest aligned neighbour approach

Whilst the verbs and other groups are processed with DL approach, HWGs with categories such as proper nouns, city, job-title, location, and country are processed with TS approach. HWGs such as number, day-unit, date-unit, month-unit, auxiliary, pronoun and postpositions, where the expected English words are specified, are processed with EEW approach. Sometimes the combination of DL and TS is also used to identify the proper alignment. At the end, nearest aligned neighbour approach is used to align the unaligned HWGs.

Dictionary Lookup

The corpus we used in our experiments is encoded in Unicode and therefore the word matching process requires dictionary entries to be in Unicode encoding. The only English-Hindi dictionary we found is called, "**shabdakoSha"** and is freely available from (WWW2). In this dictionary, the ITRANS transliteration system is followed, i.e. Hindi entries are not written in the Devanagari script, but in the Roman script. This dictionary has around 15,000 English words, each with an average of 4 relevant Hindi words. Following

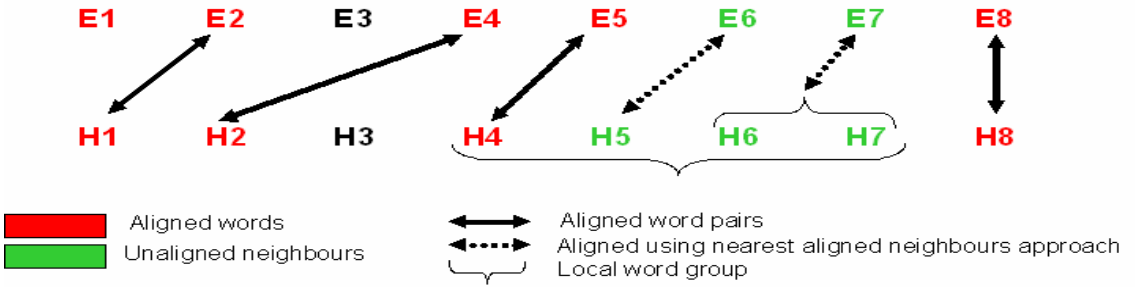


Figure 3.2 Nearest Aligned Neighbours Approach

ITRANS conventions, a parser was developed to convert all these entries into Unicode. Given a set of English and Hindi words, the algorithm presented in figure 3.1 is executed to search for the best translation among the English words.

Transliteration Similarity

A transliteration system maintains a consistent correspondence between the alphabets of two languages, irrespective of sound (Manning & Schütze, 2003). Given two words, each from a different language, we define "transliteration similarity" as the measure of likeness between them. This could exist due to the word in one language being inherited or adopted by the other language, or because the word is a proper noun. Named entities such as city, job-title, location, country and proper nouns, all recognized by the local word grouping algorithm are compared using a transliteration similarity approach. This likeness is counted using a table that lists letter correspondences between the alphabets of two languages. For the English and Hindi languages, it is possible to come up with a table that defines letter correspondence between the alphabets of two languages. For example,

A → अ, B → ब, Bh → भ, Ch → च,

D → द, Dh → ध and so on…

A bidirectional mapping is established between each character in the English and Hindi alphabets. When DL is not able to find any specific English word in dictionary, this approach is used to find the transliteration similarity between the unaligned words. Sometimes because the words in a Hindi sentence are not spelled correctly, when DL issues a query to dictionary, none of the Hindi words appearing in a Hindi sentence match with the words returned from dictionary. We use a dynamic programming algorithm "edit-distance" to calculate similarity between these words (WWW3). According to WWW3, *"The edit distance of two strings, s1 and s2, is defined as the* ***minimum*** *number of* ***point mutations*** *required to change s1 into s2, where a point mutation is one of: change a letter, insert a letter or delete a letter."* The lower the distance, the greater the similarity. From our experiments of 100 proper noun pairs, we found that if the similarity is greater than 75%, the words can be reliably aligned with each other. We consider a pair with the highest similarity. E.g.: **Aswani → अासवानी.** Here we remove vowels in both strings, except those that appear at the start of words. After the removal of vowels from the English and Hindi texts, the resulting text would be: **Aswn → असवन.** The Hindi text is then converted into English text using the transliteration table: **Aswn → Aswn.** The two texts are then compared using an "edit-distance" algorithm.

Expected English word(s)

For HWGs which are categorised as numbers, job-titles or postpositions, it is possible to specify the expected English word or words that can be found in the parallel English text. The algorithm retrieves expected English word(s) from the HWGs and tries to locate them in the English sentence. This approach can be useful to locate one or more English words that align with one or more Hindi words. For example, the number "बयालिस" whose equivalent translation in English is "forty two" has two words in English, and the postposition "के सामने", whose equivalent translation in English is "in front of", has three words in English. These are examples of many-to-many word alignment.

Nearest Aligned Neighbours

At the end of the first three stages of the word alignment process, many words remain unaligned. Here we introduce a new approach, called the "Nearest Aligned Neighbours approach". In certain cases, words in English-Hindi phrases follow a similar order. The Nearest Aligned Neighbours approach works on this principle and aligns one or more words with one of the English words. A local word grouping algorithm, explained in section 3.1, groups such phrases and tags them as "group". Considering one HWG at a time, we find the nearest Hindi word that is already aligned with one or more English word(s). We assume that the words in English-Hindi phrases follow a similar order and align the rest words in that group accordingly. An example of alignment using the Nearest Aligned Neighbours approach is given in Figure 3.2. Word H4 is already aligned with E5, and H3, H5, H6 and H7 are yet to be aligned. The local word grouping algorithm has tagged a sequence of H4, H5, H6 and H7 as a single group. At the same time, H6 and H7 are also grouped as a single group. The algorithm searches for the aligned Hindi word, which, in this case, is H4 and aligns H5 with E6 and the group of H6 and H7 with E7.

4 Results

<EnglishSentence>A fair deal and prosperity go hand in hand
<HindiSentence>एक अच्छा सौदा और समृद्धि साथ-साथ चलते हैं ।<
<EnglishWord>A</EnglishWord>
<HindiWord>एक</HindiWord>
<EnglishWord>fair</EnglishWord>
<HindiWord>अच्छा</HindiWord>
<EnglishWord>deal</EnglishWord>
<HindiWord>सौदा</HindiWord>
<EnglishWord>and</EnglishWord>
<HindiWord>और</HindiWord>
<EnglishWord>prosperity</EnglishWord>
<HindiWord>समृद्धि</HindiWord>
<EnglishWord>hand in hand</EnglishWord>
<HindiWord>साथ साथ</HindiWord>
<EnglishWord>go</EnglishWord>
<HindiWord>चलते हैं</HindiWord>

Figure 4.1 Word Alignment Results

We performed manual evaluation of our word alignment algorithm on a set of parallel data aligned at the sentence level. The parallel texts consist of 3954 English and 5361 Hindi words taken from the EMILLE Corpus. We calculate our results in terms of the number of aligned English word groups. The precision is calculated as the ratio of the number of correctly aligned English word groups to the total number of English word groups aligned by the system, and recall is calculated as the ratio of the number of correctly aligned English word groups to the total number of English word groups created by the system. We obtained 77% precision and 67.79% recall for many-to-many word alignment. Figure 4.1 shows an example of the word alignment results.

5 Future works

It would be useful to evaluate separate stages (i.e. DL, TS, EEW and Nearest Aligned Neighbours approach) in the word alignment algorithm separately. We aim to do this as part of a failure analysis of the algorithm in future. We also aim to improve our alignment results by using Part-of-Speech information for the English texts. We aim to implement or use local word grouping rules for the English text and improve our existing word grouping rules for the Hindi texts. The Nearest Aligned Neighbours approach suggests possible alignments, but we are trying to integrate some statistical ranking algorithms in order to suggest more reliable pairs of alignment. Yarowsky et al. (2001) introduced a new method for developing a Part-of-Speech tagger by projecting tags across aligned corpora. They used this technique to supply data for a supervised learning technique to acquire a French part-of-speech tagger. We aim to use our English-Hindi word alignment results to bootstrap a Part-of-Speech tagger for the Hindi language.

References

Bal Anand, 2001, *Hindi Grammar Books for standard 5 to standard 10*, Navneet Press, India.

Baker P., Bontcheva K., Cunningham H., Gaizauskas R., Hamza O., Hardie A., Jayaram B.D., Leisher M., McEnery A.M., Maynard D., Tablan V., Ursu C., Xiao Z., 2004, *Corpus linguistics and South Asian languages: Corpus creation and tool development,* Literary and Linguistic Computing, 19(4), pp. 509-524.

Brown, P., Lai, J. C., and Mercer, R., 1991, *Aligning Sentences in Parallel Corpora*, In Proceedings of ACL-91, Berkeley CA.

Chen S., 1993, *Aligning sentences in bilingual corpora using lexical information*, Proceedings of the 31st conference on Association for Computational Linguistics**,** pp. 9 – 16, Columbus, Ohio.

Gale W., and Church K., 1993, *A program for aligning sentences in bilingual corpora,* Proceedings of the 29th conference of the Association for Computational Linguistics, pp.177-184, June 18-21, 1991, Berkeley, California.

Haruno M. and Yamazaki T., 1996, *High-performance bilingual text alignment using statistical and dictionary information*, Proceedings of the 34th conference of the Association for Computational Linguistics, pp. 131 – 138, Santa Cruz, California.

Kay M. and Roscheisen M., 1993, *Text translation alignment*, Computational Linguistics, 19(1):75--102.

Manning C. and Schütze H., 2003, *Foundations of Statistical Natural Language Processing,* MIT Press, Cambridge, Massachusetts.

Mark D., 2004, *Technical Report on Unicode Standard Annex #29 - Text Boundaries*, Version 4.0.1, Unicode Inc., http://www.unicode.org/reports/tr29/ [22/11/04].

Mayers A., Grishman R., Kosaka M., 1998, *A Multilingual Procedure for Dictionary-Based Sentence Alignment*, Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup.

Maynard D., Tablan V., Bontcheva K., Cunningham H., 2003, *Rapid customisation of an Information Extraction system for surprise languages,* ACM Transactions on Asian Language Information Processing, Special issue on Rapid Development of Language Capabilities: The Surprise Languages.

Ray, P, Harish V., Sarkar, S., and Basu, A., 2003, *Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi*, Proceedings of the 1st International Conference on Natural Language Processing (ICON 2003); Mysore.

Simard M. and Pierre P., 1996, *Bilingual Sentence Alignment: Balancing Robustness and Accuracy*, Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA-96), pp. 135-144, Montreal, Quebec, Canada.

Ta A., 2002, *A Door into Hindi*, NC State University, http://www.ncsu.edu/project/hindi_lessons/lessons.html [22/11/04]

Warwick S., Catizone, R., and Graham R., 1989, *Deriving Translation Data from Bilingual Texts*, in Proceedings of the First International Lexical Acquisition Workshop, Detroit.

WU D., Jul 2000, *Alignment,* In Robert DALE, Hermann MOISL, and Harold SOMERS (editors)*,* Handbook of Natural Language Processing. pp. 415-458. New York: Marcel Dekker. ISBN 0-8247-9000-6.

WWW1, *Devanagari Unicode Chart, the Unicode Standard*, Version 4.0, Unicode Inc.,http://www.unicode.org/charts/PDF/U0900.pdf [22/03/05].

WWW2, *English-Hindi dictionary source*, http://sanskrit.gde.to/hindi/dict/eng-hin_guj.itx [22/03/05].

WWW3, *Dynamic Programming Algorithm (DPA) for Edit-Distance*, http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/ [22/03/05]

Yarowsky, D., G. Ngai and R. Wicentowski, 2001, *Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora,* In Proceedings of HLT 2001, First International Conference on Human Language Technology Research.

Word Alignment for Languages with Scarce Resources

Joel Martin
National Research Council
Ottawa, ON, K1A 0R6
Joel.Martin@cnrc-nrc.gc.ca

Rada Mihalcea
University of North Texas
Denton, TX 76203
rada@cs.unt.edu

Ted Pedersen
University of Minnesota
Duluth, MN 55812
tpederse@umn.edu

Abstract

This paper presents the task definition, resources, participating systems, and comparative results for the shared task on word alignment, which was organized as part of the ACL 2005 Workshop on Building and Using Parallel Texts. The shared task included English–Inuktitut, Romanian–English, and English–Hindi sub-tasks, and drew the participation of ten teams from around the world with a total of 50 systems.

1 Defining a Word Alignment Shared Task

The task of word alignment consists of finding correspondences between words and phrases in parallel texts. Assuming a sentence aligned bilingual corpus in languages L1 and L2, the task of a word alignment system is to indicate which word token in the corpus of language L1 corresponds to which word token in the corpus of language L2.

This year's shared task follows on the success of the previous word alignment evaluation that was organized during the HLT/NAACL 2003 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond" (Mihalcea and Pedersen, 2003). However, the current edition is distinct in that it has a focus on languages with scarce resources. Participating teams were provided with training and test data for three language pairs, accounting for different levels of data scarceness: (1) *English–Inuktitut* (2 million words training data), (2) *Romanian–English* (1 million words), and (3) *English–Hindi* (60,000 words).

Similar to the previous word alignment evaluation and with the Machine Translation evaluation exercises organized by NIST, two different subtasks were defined: (1) *Limited resources*, where systems were allowed to use only the resources provided. (2) *Unlimited resources*, where systems were allowed to use any resources in addition to those provided. Such resources had to be explicitly mentioned in the system description.

Test data were released one week prior to the deadline for result submissions. Participating teams were asked to produce word alignments, following a common format as specified below, and submit their output by a certain deadline. Results were returned to each team within three days of submission.

1.1 Word Alignment Output Format

The word alignment result files had to include one line for each word-to-word alignment. Additionally, they had to follow the format specified in Figure 1. Note that the $S|P$ and confidence fields overlap in their meaning. The intent of having both fields available was to enable participating teams to draw their own line on what they considered to be a Sure or Probable alignment. Both these fields were optional, with some standard values assigned by default.

1.1.1 A Running Word Alignment Example

Consider the following two aligned sentences:
[English] *<s snum=18> They had gone . </s>*
[French] *<s snum=18> Ils étaient allés . </s>*
A correct word alignment for this sentence is:

```
18 1 1
18 2 2
18 3 3
18 4 4
```


Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 65–74,
Ann Arbor, June 2005.

sentence_no position_L1 position_L2 [$S|P$] [confidence]

where:
∘ **sentence_no** represents the id of the sentence within the test file. Sentences in the test data already have an id assigned. (see the examples below)
∘ **position_L1** represents the position of the token that is aligned from the text in language L1; the first token in each sentence is token 1. (not 0)
∘ **position_L2** represents the position of the token that is aligned from the text in language L2; again, the first token is token 1.
∘ **S|P** can be either S or P, representing a Sure or Probable alignment. All alignments that are tagged as S are also considered to be part of the P alignments set (that is, all alignments that are considered "Sure" alignments are also part of the "Probable" alignments set). If the $S|P$ field is missing, a value of S will be assumed by default.
∘ **confidence** is a real number, in the range (0-1] (1 meaning highly confident, 0 meaning not confident); this field is optional, and by default confidence number of 1 was assumed.

Figure 1: Word Alignment file format

stating that: all the word alignments pertain to sentence 18, the English token 1 *They* aligns with the French token 1 *Ils*, the English token 2 *had* aligns with the French token 2 *étaient*, and so on. Note that punctuation is also aligned (English token 4 aligned with French token 4), and counts toward the final evaluation figures.

Alternatively, systems could also provide an $S|P$ marker and/or a confidence score, as shown in the following example:

```
18 1 1 1
18 2 2 P 0.7
18 3 3 S
18 4 4 S 1
```

with missing $S|P$ fields considered by default S, and missing confidence scores considered by default 1.

1.2 Annotation Guide for Word Alignments

The word alignment annotation guidelines are similar to those used in the 2003 evaluation.

1. All items separated by a white space are considered to be a word (or token), and therefore have to be aligned (punctuation included).
2. Omissions in translation use the NULL token, i.e. token with id 0.
3. Phrasal correspondences produce multiple word-to-word alignments.

2 Resources

The shared task included three different language pairs, accounting for different language and data characteristics. Specifically, the three subtasks addressed the alignment of words in English–Inuktitut, Romanian–English, and English–Hindi parallel texts. For each language pair, training data were provided to participants. Systems relying only on these resources were considered part of the *Limited Resources* subtask. Systems making use of any additional resources (e.g. bilingual dictionaries, additional parallel corpora, and others) were classified under the *Unlimited Resources* category.

2.1 Training Data

Three sets of training data were made available. All data sets were sentence-aligned, and pre-processed (i.e. tokenized and lower-cased), with identical preprocessing procedures used for training, trial, and test data.

English–Inuktitut. A collection of sentence-aligned English–Inuktitut parallel texts from the Legislative Assembly of Nunavut (Martin et al., 2003). This collection consists of approximately 2 million Inuktitut tokens (1.6 million words) and 4 million English tokens (3.4 million words). The Inuktitut data was originally encoded in Unicode representing a syllabics orthography (qaniujaaqpait), but was transliterated to an ASCII encoding of the standardized roman orthography (qaliujaaqpait) for this evaluation.

Romanian–English. A set of Romanian–English parallel texts, consisting of about 1 million Romanian words, and about the same number of English words. This is the same training data set as used in the 2003 word alignment evaluation (Mihalcea and Pedersen, 2003). The data consists of:

- Parallel texts collected from the Web using a semi-supervised approach. The URLs format for pages containing potential parallel translations were manually identified (mainly from the archives of Romanian newspapers). Next, texts were automatically downloaded and sentence aligned. A manual verification of the alignment was also performed. These data collection process resulted in a corpus of about 850,000 Romanian words, and about 900,000 English words.

- Orwell's 1984, aligned within the MULTEXT-EAST project (Erjavec et al., 1997), with about 130,000 Romanian words, and a similar number of English words.

- The Romanian Constitution, for about 13,000 Romanian words and 13,000 English words.

English–Hindi. A collection of sentence aligned English–Hindi parallel texts, from the Emille project (Baker et al., 2004), consisting of approximately English 60,000 words and about 70,000 Hindi words. The Hindi data was encoded in Unicode Devangari script, and used the UTF–8 encoding. The English–Hindi data were provided by Niraj Aswani and Robert Gaizauskas from University of Sheffield (Aswani and Gaizauskas, 2005b).

2.2 Trial Data

Three sets of trial data were made available at the same time training data became available. Trial sets consisted of sentence aligned texts, provided together with manually determined word alignments. The main purpose of these data was to enable participants to better understand the format required for the word alignment result files. For some systems, the trial data has also played the role of a validation data set used for system parameter tuning. Trial sets consisted of 25 English–Inuktitut and English–Hindi aligned sentences, and a larger set of 248 Romanian–English aligned sentences (the same as the test data used in the 2003 word alignment evaluation).

2.3 Test Data

A total of 75 English–Inuktitut, 90 English–Hindi, and 200 Romanian–English aligned sentences were released one week prior to the deadline. Participants were required to run their word alignment systems on one or more of these data sets, and submit word alignments. Teams were allowed to submit an unlimited number of results sets for each language pair.

2.3.1 Gold Standard Word Aligned Data

The gold standard for the three language pair alignments were produced using slightly different alignment procedures.

For English–Inuktitut, annotators were instructed to align Inuktitut words or phrases with English phrases. Their goal was to identify the smallest phrases that permit one-to-one alignments between English and Inuktitut. These phrase alignments were converted into word-to-word alignments in the following manner. If the aligned English and Inuktitut phrases each consisted of a single word, that word pair was assigned a Sure alignment. Otherwise, all possible word-pairs for the aligned English and Inuktitut phrases were assigned a Probable alignment. Disagreements between the two annotators were decided by discussion.

For Romanian–English and English–Hindi, annotators were instructed to assign an alignment to *all* words, with specific instructions as to when to assign a NULL alignment. Annotators were not asked to assign a Sure or Probable label. Instead, we had an arbitration phase, where a third annotator judged the cases where the first two annotators disagreed. Since an inter-annotator agreement was reached for all word alignments, the final resulting alignments were considered to be Sure alignments.

3 Evaluation Measures

Evaluations were performed with respect to four different measures. Three of them – precision, recall, and F-measure – represent traditional measures in Information Retrieval, and were also frequently used in previous word alignment literature. The fourth measure was originally introduced by (Och and Ney, 2000), and proposes the notion of *quality of word alignment*.

Given an alignment $\mathcal{A}$, and a gold standard alignment $\mathcal{G}$, each such alignment set eventually consisting of two sets $\mathcal{A}_\mathcal{S}$, $\mathcal{A}_\mathcal{P}$, and $\mathcal{G}_\mathcal{S}$, $\mathcal{G}_\mathcal{P}$ corresponding to Sure and Probable alignments, the following measures are defined (where T is the alignment type, and can be set to either S or P).

$$P_T = \frac{|A_T \cap G_T|}{|A_T|} \quad (1)$$

$$R_T = \frac{|A_T \cap G_T|}{|G_T|} \quad (2)$$

$$F_T = \frac{2P_T R_T}{P_T + R_T} \quad (3)$$

$$AER = 1 - \frac{|A_P \cap G_S| + |A_P \cap G_P|}{|A_P| + |G_S|} \quad (4)$$

Each word alignment submission was evaluated in terms of the above measures. Given numerous (constructive) debates held during the previous word alignment evaluation, which questioned the informativeness of the NULL alignment evaluations, we decided

Team	System name	Description
Carnegie Mellon University	SPA	(Brown et al., 2005)
Information Sciences Institute / USC	ISI	(Fraser and Marcu, 2005)
Johns Hopkins University	JHU	(Schafer and Drabek, 2005)
Microsoft Research	MSR	(Moore, 2005)
Romanian Academy Institute of Artificial Intelligence	TREQ-AL, MEBA, COWAL	(Tufis et al., 2005)
University of Maryland / UMIACS	UMIACS	(Lopez and Resnik, 2005)
University of Sheffield	Sheffield	(Aswani and Gaizauskas, 2005a)
University of Montreal	JAPA, NUKTI	(Langlais et al., 2005)
University of Sao Paulo, University of Alicante	LIHLA	(Caseli et al., 2005)
University Jaume I	MAR	(Vilar, 2005)

Table 1: Teams participating in the word alignment shared task

to evaluate only no-NULL alignments, and thus the NULL alignments were removed from both submissions and gold standard data. We conducted therefore 7 evaluations for each submission file: AER, Sure/Probable Precision, Sure/Probable Recall, and Sure/Probable F-measure, all of them measured on no-NULL alignments.

4 Participating Systems

Ten teams from around the world participated in the word alignment shared task. Table 1 lists the names of the participating systems, the corresponding institutions, and references to papers in this volume that provide detailed descriptions of the systems and additional analysis of their results.

Seven teams participated in the Romanian–English subtask, four teams participated in the English–Inuktitut subtask, and two teams participated in the English–Hindi subtask. There were no restrictions placed on the number of submissions each team could make. This resulted in a total of 50 submissions from the ten teams, where 37 sets of results were submitted for the Romanian–English subtask, 10 for the English–Inuktitut subtask, and 3 for the English–Hindi subtask. Of the 50 total submissions, there were 45 in the *Limited resources* subtask, and 5 in the *Unlimited resources* subtask. Tables 2, 4 and 6 show all of the submissions for each team in the three subtasks, and provide a brief description of their approaches.

Results for all participating systems, including precision, recall, F-measure, and alignment error rate are listed in Tables 3, 5 and 7. Ranked results for all systems are plotted in Figures 2, 3 and 4. In the graphs, systems are ordered based on their AER scores. System names are preceded by a marker to indicate the system type: L stands for *Limited Resources*, and U stands for *Unlimited Resources*.

While each participating system was unique, there were a few unifying themes. Several teams had approaches that relied (to varying degrees) on an IBM model of statistical machine translation (Brown et al., 1993), with different improvements brought by different teams, consisting of new submodels, improvements in the HMM model, model combination for optimal alignment, etc. Se-veral teams used symmetrization metrics, as introduced in (Och and Ney, 2003) (union, intersection, refined), most of the times applied on the alignments produced for the two directions source–target and target–source, but also as a way to combine different word alignment systems. Significant improvements with respect to baseline word alignment systems were observed when the vocabulary was reduced using simple stemming techniques, which seems to be a particularly effective technique given the data sparseness problems associated with the relatively small amounts of training data.

In the *unlimited resources* subtask, systems made use of bilingual dictionaries, human–contributed word alignments, or syntactic constraints derived from a dependency parse tree applied on the English side of the corpus.

When only small amounts of parallel corpora were available (i.e. the English–Hindi subtask), the use of additional resources resulted in absolute improvements of up to 20% as compared to the case when the word alignment systems were based exclusively on the parallel texts. Interestingly, this was not the case for the language pairs that had larger training corpora (i.e. Romanian–English, English–Inuktitut), where the *limited resources* systems seemed to lead to comparable or sometime even better results than those that relied on *unlimited resources*. This suggests

that the use of additional resources does not seem to contribute to improvements in word alignment quality when enough parallel corpora are available, but they can make a big difference when only small amounts of parallel texts are available.

Finally, in a comparison across language pairs, the best results are obtained in the English–Inuktitut task, followed by Romanian–English, and by English–Hindi, which corresponds to the ordering of the sizes of the training data sets. This is not surprising since, like many other NLP tasks, word alignment seems to highly benefit from large amounts of training data, and thus better results are obtained when larger training data sets are available.

5 Conclusion

A shared task on word alignment was organized as part of the ACL 2005 Workshop on Building and Using Parallel Texts. The focus of the task was on languages with scarce resources, with evaluations of alignments for three different language pairs: English–Inuktitut, English–Hindi, and Romanian–English. The task drew the participation of ten teams from around the world, with a total of 50 systems. In this paper, we presented the task definition, resources involved, and shortly described the participating systems. Comparative evaluations of results led to insights regarding the development of word alignment algorithms for languages with scarce resources, with performance evaluations of (1) various algorithms, (2) different amounts of training data, and (3) different additional resources. Data and evaluation software used in this exercise are available online at http://www.cs.unt.edu/~rada/wpt05.

Acknowledgments

There are many people who contributed greatly to making this word alignment evaluation task possible. We are grateful to all the participants in the shared task, for their hard work and involvement in this evaluation exercise. Without them, all these comparative analyses of word alignment techniques would not be possible. In particular, we would like to thank Dan Tufiş and Bob Moore for their helpful comments concerning the Romanian–English data. We would also like to thank Benoit Farley for his valuable assistance with the English–Inuktitut data.

We are very thankful to Niraj Aswani and Rob Gaizauskas from University of Sheffield for making possible the English–Hindi word alignment evaluation. They provided sentence aligned training data from the Emille project, as well as word aligned trial and test data sets.

We are also grateful to all the Program Committee members for their comments and suggestions, which helped us improve the definition of this shared task.

References

N. Aswani and R. Gaizauskas. 2005a. Aligning words in english-hindi parallel corpora. In *(this volume)*.

N. Aswani and R. Gaizauskas. 2005b. A hybrid approach to align sentences and words in English-Hindi parallel corpora. In *Proceedings of the ACL Workshop on "Building and Exploiting Parallel Texts"*, Ann Arbor, MI.

P. Baker, K. Bontcheva, H. Cunningham, R. Gaizauskas, O. Hamza, A. Hardie, B. Jayaram, M. Leisher, A McEnery, D Maynard, V. Tablan, C. Ursu, and Z. Xiao. 2004. Corpus linguistics and south asian languages: Corpus creation and tool development. *Literary and Linguistic Computing*, 19(4).

P. Brown, S. della Pietra, V. della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2).

R. D. Brown, J.D. Kim, P. J. Jansen, and J. G. Carbonell. 2005. Symmetric probabilistic alignment. In *(this volume)*.

H. Caseli, M. G. V. Nunes, and M. L. Forcada. 2005. Lihla: Shared task system description. In *(this volume)*.

T. Erjavec, N. Ide, and D. Tufis. 1997. Encoding and parallel alignment of linguistic corpora in six central and Eastern European languages. In *Proceedings of the Joint ACH/ALL Conference*, Queen's University, Kingston, Ontario, June.

A. Fraser and D. Marcu. 2005. Isi's participation in the romanian-english alignment task. In *(this volume)*.

P. Langlais, F. Gotti, and G. Cao. 2005. Nukti: English-inuktitut word alignment system description. In *(this volume)*.

A. Lopez and P. Resnik. 2005. Improved hmm alignment models for languages with scarce resources. In *(this volume)*.

J. Martin, H. Johnson, B. Farley, and A. Maclachlan. 2003. Aligning and using an english-inuktitut parallel corpus. In *Proceedings of the HLT-NAACL Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada.

R. Mihalcea and T. Pedersen. 2003. An evaluation exercise for word alignment. In *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada, May.

R. Moore. 2005. Association-based bilingual word alignment. In *(this volume)*.

F. Och and H. Ney. 2000. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrucken, Germany, August.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

C. Schafer and E. Drabek. 2005. Models for inuktitut-english word alignment. In *(this volume)*.

D. Tufis, R. Ion, A. Ceausu, and D. Stefanescu. 2005. Combined word alignments. In *(this volume)*.

J.M. Vilar. 2005. Experiments using mar for aligning corpora. In *(this volume)*.

System	Resources	Description
JHU.AER.Emphasis.I	Limited	A word alignment system optimized for the characteristics of English–Inuktitut, exploiting cross-lingual affinities at sublexical level and regular patterns of transliteration. The system is based on classifier combination, performed under an AER target evaluation metric.
JHU.AER.Emphasis.II	Limited	Same as JHU.AER.Emphasis.I, but with a different minimum required votes for classifier combination.
JHU.F-meas.Emphasis	Limited	Same as JHU.AER.Emphasis.I, with classifier combination performed under an F-measure target evaluation metric.
JHU.AER.F-meas.AER DualEmphasis	Limited	Same as JHU.AER.Emphasis.I, with a dual emphasis on AER and F-measure.
JHU.Recall.Emphasis	Limited	Same as JHU.AER.Emphasis.I, with an emphasis on recall.
LIHLA	Limited	A word alignment tool based on language-independent heuristics. Starts with two bilingual probabilistic lexicons (source-target and target-source) generated by NATools (http://natura.di.uminho.pt/natura/natura/), which are combined with some language-independent heuristics that try to find the best alignment.
UMIACS.limited	Limited	A system using IBM Model 4 with improvements brought in the HMM model.
UMontreal.NUKTI	Limited	A system based on computation of log-likelihood ratios between all Inuktitut substrings and English words. Alignment with a greedy strategy trying to optimize this association score.
UMontreal.Japa-cart	Limited	A system based on alignment with a sentence aligner where Inuktitut and English words are considered to be sentences. In case a n-m alignment is produced, its cartesian product is output as the final alignment.
UMontreal.Japa-nukti	Limited	Same as UMontreal.Japa-cart except for the treatment of the n-m pairs (n,m $\geq$ 1). Instead of generating the cartesian product, this method uses the NUKTI approach to figure out which words should be aligned.

Table 2: Short description for English–Inuktitut systems

System	P_S	R_S	F_S	P_P	R_P	F_P	AER
Limited Resources							
JHU.AER.Emphasis.II	34.19%	76.79%	47.32%	**96.66%**	32.35%	48.37%	**9.46%**
JHU.AER.Emphasis.I	28.15%	82.25%	41.95%	90.65%	39.35%	54.88%	11.49%
JHU.F-measure.AER.DualEmphasis	19.71%	92.15%	32.47%	84.38%	58.62%	69.18%	14.25%
UMIACS.limited	**49.86%**	62.80%	55.59%	89.16%	16.68%	28.11%	22.51%
LIHLA	46.55%	73.72%	**57.07%**	79.53%	18.71%	30.30%	22.72%
JHU.F-measure.Emphasis	13.06%	91.81%	22.87%	70.67%	73.78%	**72.19%**	26.70%
UMontreal.nukti	12.24%	86.01%	21.43%	63.09%	65.87%	64.45%	34.06%
JHU.Recall.Emphasis	10.68%	**93.86%**	19.18%	62.63%	**81.74%**	70.92%	34.18%
UMontreal.Japa-nukti	9.62%	67.58%	16.84%	51.34%	53.60%	52.44%	46.64%
UMontreal.Japa-cart	0.00%	0.00%	0.00%	26.17%	74.49%	38.73%	71.27%

Table 3: Results for English–Inuktitut

System	Resources	Description
CMU.SPA contiguous	Limited	A tool based on Symmetric Probabilistic Alignment (SPA), which maximizes bi-directional translation probabilities of words in a selected source-language n-gram and every possible target-language n-gram. Probabilities are derived from a pair of probabilistic lexicons (source-to-target and target-to-source). Only contiguous target-language n-grams are considered as possible alignments.
CMU.SPA non-contiguous	Limited	Same as CMU.SPA.contiguous, but both contiguous and non-contiguous target-language n-grams are considered as possible alignments
CMU.SPA human-augmented	Unlimited	Same as CMU.SPA.contiguous, but the probabilistic dictionaries were modified with word and phrasal translations extracted from a human alignment of 204 sentences in the training corpus.
ISI.RUN1	Limited	A baseline word-based system using IBM Model 4 as implemented in Giza++. Different subruns include the two separate direction En–Ro, Ro–En, as well as the "union", "intersection", and "refined" symmetrization metrics, as defined in (Och and Ney, 2003)
ISI.RUN2	Limited	Same as ISI.RUN1, but uses stems of size 4 (instead of words) for both English and Romanian.
ISI.RUN4	Limited	A system using IBM Model 4 and a new submodel based on the intersection of two starting alignments. The submodels are grouped into a log-linear model, with optimal weights found through a search algorithm.
ISI.RUN5	Limited	Same as ISI.RUN4, but with 5 additional submodels, using translation tables for En–Ro, Ro–En, backoff fertility, zero or non-zero fertility English word penalty
UJaume.MAR	Limited	A new alignment model based on a recursive approach. Due to its high computational cost, heuristics have been used to split training and test data in smaller chunks.
USaoPaulo.LIHLA	Limited	A word alignment tool based on language-independent heuristics. Starts with two bilingual probabilistic lexicons (source-target and target-source) generated by NATools (http://natura.di.uminho.pt/natura/natura/), which are combined with some language-independent heuristics that try to find the best alignment.
MSR.word-align	Limited	A system based on competitive linking, first by log-likelihood-ratio association score, then by probability of link given joint occurrence; constrained by measuring monontonicity of alignment, and augmented with 1-2 and 2-1 alignments also derived by competitive linking.
RACAI.MEBA-V1	Limited	A system based on GIZA++, with a translation model constructed using seven major parameters that control the contribution of various heuristics (cognates, relative distance, fertility, displacement, etc.)
RACAI.MEBA-V2	Limited	Same as RACAI.MEBA-V1, but with a different set of parameters.
RACAI.TREQ-AL	Unlimited	Same as RACAI.MEBA-V1, but with an additional resource consisting of a translation dictionary extracted from the alignment of the Romanian and English WordNet.
RACAI.COWAL	Unlimited	A combination (union) of RACAI.MEBA and RACAI.TREQ-AL.
UMIACS.limited	Limited	A system using IBM Model 4 with improvements brought in the HMM model.
UMIACS.unlimited	Unlimited	Same as UMIACS.limited, but also integrating a distortion model based on a dependency parse built on the English side of the parallel corpus.

Table 4: Short description for Romanian–English systems

System	P_S	R_S	F_S	P_P	R_P	F_P	AER
Limited Resources							
ISI.Run5.vocab.grow	87.90%	63.08%	**73.45%**	87.90%	63.08%	**73.45%**	**26.55%**
ISI.Run3.vocab.grow	87.93%	62.98%	73.40%	87.93%	62.98%	73.40%	26.60%
ISI.Run4.vocab.grow	88.31%	62.75%	73.37%	88.31%	62.75%	73.37%	26.63%
ISI.Run2.vocab.grow	81.84%	66.28%	73.25%	81.84%	66.28%	73.25%	26.75%
ISI.Run5.simple.union	81.78%	65.35%	72.64%	81.78%	65.35%	72.64%	27.36%
ISI.Run5.simple.normal	87.09%	61.93%	72.39%	87.09%	61.93%	72.39%	27.61%
ISI.Run4.simple.union	81.85%	64.69%	72.27%	81.85%	64.69%	72.27%	27.73%
ISI.Run5.simple.inverse	86.96%	61.75%	72.22%	86.96%	61.75%	72.22%	27.78%
ISI.Run3.simple.normal	87.11%	61.63%	72.19%	87.11%	61.63%	72.19%	27.81%
ISI.Run3.simple.union	81.00%	65.05%	72.15%	81.00%	65.05%	72.15%	27.85%
ISI.Run4.simple.normal	87.20%	61.34%	72.02%	87.20%	61.34%	72.02%	27.98%
ISI.Run5.simple.intersect	93.77%	58.33%	71.93%	93.77%	58.33%	71.93%	28.07%
ISI.Run3.simple.intersect	93.92%	57.96%	71.68%	93.92%	57.96%	71.68%	28.32%
ISI.Run3.simple.inverse	86.12%	61.37%	71.67%	86.12%	61.37%	71.67%	28.33%
ISI.Run4.simple.inverse	87.33%	60.78%	71.67%	87.33%	60.78%	71.67%	28.33%
ISI.Run4.simple.intersect	**94.29%**	57.42%	71.38%	**94.29%**	57.42%	71.38%	28.62%
ISI.Run2.simple.inverse	81.32%	63.32%	71.20%	81.32%	63.32%	71.20%	28.80%
ISI.Run2.simple.union	70.46%	**71.31%**	70.88%	70.46%	**71.31%**	70.88%	29.12%
RACAI MEBA-V1	83.21%	60.54%	70.09%	83.21%	60.54%	70.09%	29.91%
ISI.Run2.simple.intersect	94.08%	55.22%	69.59%	94.08%	55.22%	69.59%	30.41%
ISI.Run2.simple.normal	77.04%	63.20%	69.44%	77.04%	63.20%	69.44%	30.56%
RACAI MEBA-V2	77.90%	61.85%	68.96%	77.90%	61.85%	68.96%	31.04%
ISI.Run1.simple.grow	75.82%	62.23%	68.35%	75.82%	62.23%	68.35%	31.65%
UMIACS.limited	73.77%	61.69%	67.19%	73.77%	61.69%	67.19%	32.81%
ISI.Run1.simple.inverse	72.70%	57.34%	64.11%	72.70%	57.34%	64.11%	35.89%
ISI.Run1.simple.union	59.96%	68.85%	64.10%	59.96%	68.85%	64.10%	35.90%
MSR.word-align	79.54%	53.13%	63.70%	79.54%	53.13%	63.70%	36.30%
CMU.SPA.contiguous	64.96%	61.34%	63.10%	64.96%	61.34%	63.10%	36.90%
CMU.SPA.noncontiguous	64.91%	61.34%	63.07%	64.91%	61.34%	63.07%	36.93%
ISI.Run1.simple.normal	67.41%	56.81%	61.66%	67.41%	56.81%	61.66%	38.34%
ISI.Run1.simple.intersect	93.75%	45.30%	61.09%	93.75%	45.30%	61.09%	38.91%
UJaume.MAR	54.04%	64.65%	58.87%	54.04%	64.65%	58.87%	41.13%
USaoPaulo.LIHLA	57.68%	53.51%	55.51%	57.68%	53.51%	55.51%	44.49%
Unlimited Resources							
RACAI.COWAL	71.24%	**76.77%**	**73.90%**	71.24%	**76.77%**	**73.90%**	**26.10%**
RACAI.TREQ-AL	**82.08%**	60.62%	69.74%	**82.08%**	60.62%	69.74%	30.26%
UMIACS.unlimited	72.41%	**62.15%**	66.89%	72.41%	**62.15%**	66.89%	33.11%
CMU.SPA.human-augmented	64.60%	60.54%	62.50%	64.60%	60.54%	62.50%	37.50%

Table 5: Results for Romanian–English

System	Resources	Description
USheffield	Unlimited	A multi-feature approach for many-to-many word alignment. Prior to word alignment, a pattern-based local word grouping is performed for both English and Hindi. Various methods such as dictionary lookup, transliteration similarity, expected English word(s) and nearest aligned neighbors are used.
UMIACS.limited	Limited	A system using IBM Model 4 with improvements brought in the HMM model.
UMIACS.unlimited	Unlimited	Same as UMIACS.limited, but also integrating a distortion model based on a dependency parse built on the English side of the parallel corpus.

Table 6: Short description for English–Hindi systems

System	P_S	R_S	F_S	P_P	R_P	F_P	AER
Limited Resources							
UMIACS.limited	**42.90%**	**56.00%**	**48.58%**	**42.90%**	**56.00%**	**48.58%**	**51.42%**
Unlimited Resources							
USheffield	**77.03%**	**60.68%**	**67.88%**	**77.03%**	**60.68%**	**67.88%**	**32.12%**
UMIACS.unlimited	43.65%	56.14%	49.12%	43.65%	56.14%	49.12%	50.88%

Table 7: Results for English–Hindi

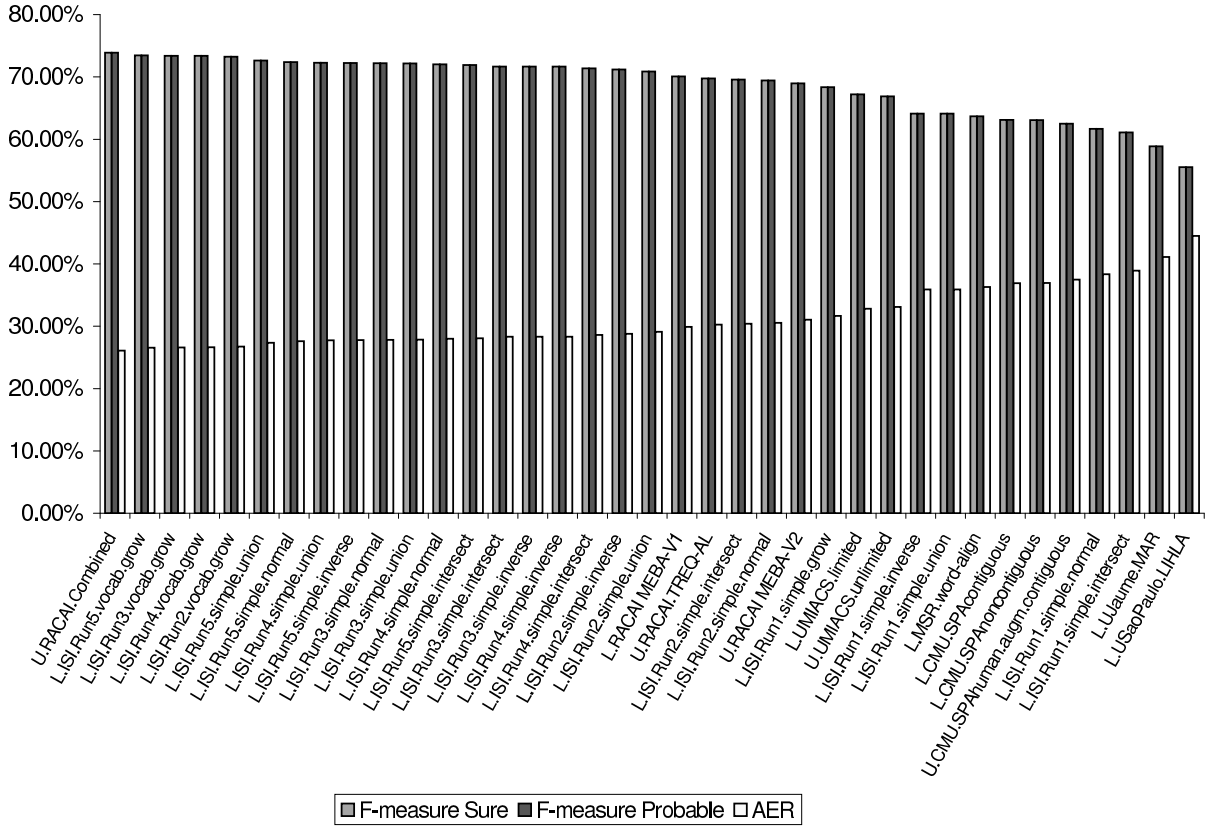


Figure 2: Ranked results for Romanian–English data

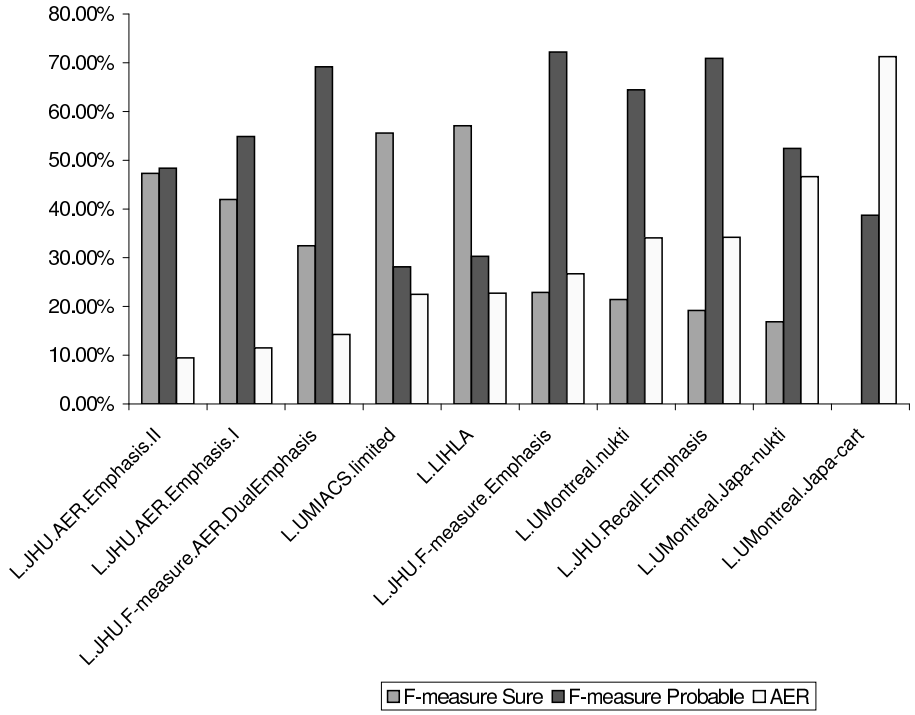


Figure 3: Ranked results for English–Inuktitut data

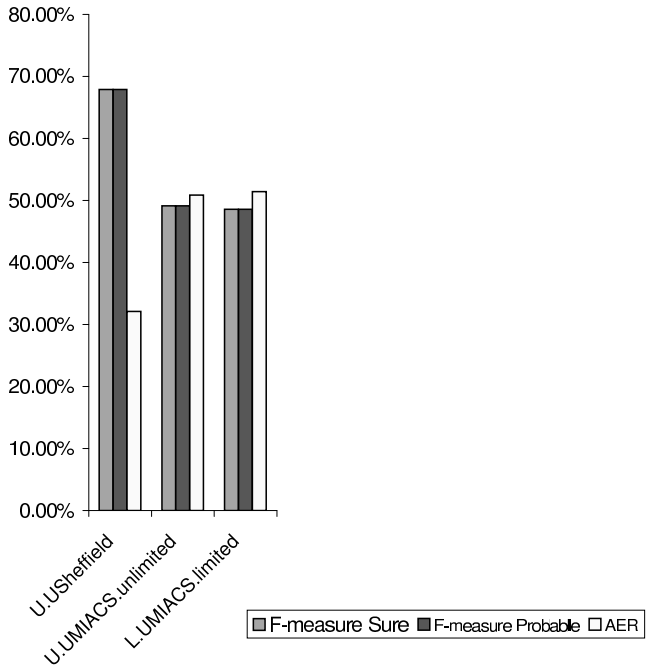


Figure 4: Ranked results for English–Hindi data

NUKTI: English-Inuktitut Word Alignment System Description

Philippe Langlais, Fabrizio Gotti, Guihong Cao
RALI
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
Succursale Centre-Ville
H3C 3J7 Montréal, Canada
http://rali.iro.umontreal.ca

Abstract

Machine Translation (MT) as well as other bilingual applications strongly rely on word alignment. Efficient alignment techniques have been proposed but are mainly evaluated on pairs of languages where the notion of word is mostly clear. We concentrated our effort on the English-Inuktitut word alignment shared task and report on two approaches we implemented and a combination of both.

1 Introduction

Word alignment is an important step in exploiting parallel corpora. When efficient techniques have been proposed (Brown et al., 1993; Och and Ney, 2003), they have been mostly evaluated on "safe" pairs of languages where the notion of word is rather clear.

We devoted two weeks to the intriguing task of aligning at the word level pairs of sentences of English and Inuktitut. We experimented with two different approaches. For the first one, we relied on an in-house sentence alignment program (JAPA) where English and Inuktitut tokens were considered as sentences. The second approach we propose takes advantage of associations computed between any English word and roughly any subsequence of Inuktitut characters seen in the training corpus. We also investigated the combination of both approaches.

2 JAPA: Word Alignment as a Sentence Alignment Task

To adjust our systems, the organizers made available to the participants a set of 25 pairs of sentences where words had been manually aligned. A fast inspection of this material reveals that in most of the cases, the alignment produced are monotonic and involve *cepts* of `n` adjacent English words aligned to a single Inuktitut word.

Many sentence alignment techniques strongly rely on the monotonic nature of the inherent alignment. Therefore, we conducted a first experiment using an in-house sentence alignment program called JAPA that we developed within the framework of the Arcade evaluation campaign (Langlais et al., 1998). The implementation details of this aligner can be found in (Langlais, 1997), but in a few words, JAPA aligns pairs of sentences by first grossly aligning their words (making use of either cognate-like tokens, or a specified bilingual dictionary). A second pass aligns the sentences in a way similar[1] to the algorithm described by Gale and Church (1993), but where the search space is constrained to be close to the one delimited by the word alignment. This technique happened to be among the most accurate of the ones tested during the Arcade exercise.

To adapt JAPA to our needs, we only did two things. First, we considered single sentences as documents, and tokens as sentences (we define a token as a sequence of characters delimited by

[1] In our case, the score we seek to globally maximize by dynamic programming is not only taking into account the length criteria described in (Gale and Church, 1993) but also a cognate-based one similar to (Simard et al., 1992).

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 75–78,
Ann Arbor, June 2005.

1-1	0.406	4-1	0.092	4-2	0.015
2-1	0.172	5-1	0.038	5-2	0.011
3-1	0.123	7-1	0.027	3-2	0.011

Table 1: The 9 most frequent English-Inuktitut patterns observed on the development set. A total of 24 different patterns have been observed.

white space). Second, since in its default setting, JAPA only considers `n-m` sentence-alignment patterns with `n,m` $\in [0, 2]$, we provided it with a new pattern distribution we computed from the development corpus (see Table 1). It is interesting to note that although English and Inuktitut have very different word systems, the length ratio (in characters) of the two sides of the TRAIN corpus is 1.05.

Each pair of documents (sentences) were then aligned separately with JAPA. `1-n` and `n-1` alignments identified by JAPA where output without further processing. Since the word alignment format of the shared task do not account directly for `n-m` alignments (`n,m` > 1) we generated the cartesian product of the two sets of words for all these `n-m` alignments produced by JAPA.

The performance of this approach is reported in Table 2. Clearly, the precision is poor. This is partly explained by the cartesian product we resorted to when `n-m` alignments were produced by JAPA. We provide in section 4 a way of improving upon this scenario.

Prec.	Rec.	F-meas.	AER
22.34	78.17	34.75	74.59

Table 2: Performance of the JAPA alignment technique on the DEV corpus.

3 NUKTI: Word and Substring Alignment

Martin et al. (2003) documented a study in building and using an English-Inuktitut bitext. They described a sentence alignment technique tuned for the specificity of the Inuktitut language, and described as well a technique for acquiring correspondent pairs of English tokens and Inuktitut substrings. The motivation behind their work was to populate a glossary with reliable such pairs. We extended this line of work in order to achieve word alignment.

3.1 Association Score

As Martin et al. (2003) pointed out, the strong agglutinative nature of Inuktitut makes it necessary to consider subunits of Inuktitut tokens. This is reflected by the large proportion of token types and hapax words observed on the Inuktitut side of the training corpus, compared to the ratios observed on the English side (see table 3).

	Inutktitut	%	English	%
tokens	2 153 034		3 992 298	
types	417 407	19.4	27 127	0.68
hapax	337 798	80.9	8 792	32.4

Table 3: Ratios of token types and happax words in the TRAIN corpus.

The main idea presented in (Martin et al., 2003) is to compute an association score between any English word seen in the training corpus and all the Inuktitut substrings of those tokens that were seen in the same region. In our case, we computed a likelihood ratio score (Dunning, 1993) for all pairs of English tokens and Inuktitut substrings of length ranging from 3 to 10 characters. A maximum of 25 000 associations were kept for each English word (the top ranked ones).

To reduce the computation load, we used a suffix tree structure and computed the association scores only for the English words belonging to the test corpus we had to align. We also filtered out Inuktitut substrings we observed less than three times in the training corpus. Altogether, it takes about one hour for a good desktop computer to produce the association scores for one hundred English words.

We normalize the association scores such that for each English word e, we have a distribution of likely Inuktitut substrings s: $\sum_s p_{llr}(s|e) = 1$.

3.2 Word Alignment Strategy

Our approach for aligning an Inuktitut sentence of K tokens I_1^K with an English sentence of N tokens E_1^N (where $K \leq N$)[2] consists of finding

[2] As a matter of fact, the number of Inuktitut words in the test corpus is always less than or equal to the number of English tokens for any sentence pair.

$K-1$ *cutting points* $c_{k\in[1,K-1]}$ ($c_k \in [1, N-1]$) on the English side. A frontier c_k delimits adjacent English words $E_{c_{k-1}+1}^{c_k}$ that are translation of the single Inuktitut word I_k. With the convention that $c_0 = 0$, $c_K = N$ and $c_{k-1} < c_k$, we can formulate our alignment problem as seeking the best word alignment $A = A(I_1^K|E_1^N)$ by maximizing:

$$A = \underset{c_1^K}{\operatorname{argmax}} \prod_{k=1}^{K} p(I_k|E_{c_{k-1}+1}^{c_k})^{\alpha_1} \times p(d_k)^{\alpha_2} \quad (1)$$

where $d_k = c_k - c_{k-1}$ is the number of English words associated to I_k; $p(d_k)$ is the prior probability that d_k English words are aligned to a single Inuktitut word, which we computed directly from Table 1; and α_1 and α_2 are two weighting coefficients.

We tried the following two approximations to compute $p(I_k|E_{c_{k-1}+1}^{c_k})$. The second one led to better results.

$$p(I_k|E_{c_{k-1}+1}^{c_k}) \simeq \begin{cases} \max_{j=c_{k-1}+1}^{c_k} p(I_k|E_j) \\ \text{or} \\ \sum_{j=c_{k-1}+1}^{c_k} p(I_k|E_j) \end{cases}$$

We considered several ways of computing the probability that an Inuktitut token I is the translation of an English one E; the best one we found being:

$$p(I|E) \simeq \sum_{s\in I} \lambda p_{llr}(s|E) + (1-\lambda) p_{ibm2}(s|E)$$

where the summation is carried over all substrings s of I of 3 characters or more. $p_{llr}(s|E)$ is the normalized log-likelihood ratio score described above and $p_{ibm2}(s|E)$ is the probability obtained from an IBM model 2 we trained after the Inuktitut side of the training corpus was segmented using a recursive procedure optimizing a frequency-based criterion. λ is a weighting coefficient.

We tried to directly embed a model trained on whole (unsegmented) Inuktitut tokens, but noticed a degradation in performance (line 2 of Table 4).

3.3 A Greedy Search Strategy

Due to its combinatorial nature, the maximization of equation 1 was barely tractable. Therefore we adopted a greedy strategy to reduce the search space. We first computed a split of the English sentence into K adjacent regions c_1^K by virtually drawing a diagonal line we would observe if a character in one language was producing a constant number of characters in the other one. An initial word alignment was then found by simply tracking this diagonal at the word granularity level.

Having this split in hand (line 1 of Table 4), we move each cutting point around its initial value starting from the leftmost cutting point and going rightward. Once a locally optimal cutting point has been found (that is, maximizing the score of equation 1), we proceed to the next one directly to its right.

3.4 Results

We report in Table 4 the performance of different variants we tried as measured on the development set. We used these performances to select the best configuration we eventually submitted.

variant	Prec.	Rec.	F-m.	AER
start (diag)	51.7	53.66	52.66	49.54
greedy (word)	61.6	63.94	62.75	35.93
greedy (best)	63.5	65.92	64.69	34.21

Table 4: Performance of several NUKTI alignment techniques measured on the DEV corpus.

It is interesting to note that the starting point of the greedy search (line 1) does better than our first approach. However, moving from this initial split clearly improves the performance (line 3). Among the greedy variants we tested, we noticed that putting much of the weight λ on the IBM model 2 yielded the best results. We also noticed that $p(d_k)$ in equation 1 did not help (α_2 was close to zero). A character-based model might have been more appropriate to the case.

4 Combination of JAPA and NUKTI

One important weakness of our first approach lies in the cartesian product we generate when JAPA produces a `n-m` ($n, m > 1$) alignment. Thus, we tried a third approach: we apply NUKTI on any `n-m` alignment JAPA produces as if this initial alignment were in fact two (small) sentences to align, `n`- and `m`-word long respectively. We can

therefore avoid the cartesian product and select word alignments more discerningly. As can be seen in Table 5, this combination improved over JAPA alone, while being worse than NUKTI alone.

5 Results

We submitted 3 variants to the organizers. The performances for each method are gathered in Table 5. The order of merit of each approach was consistent with the performance we measured on the DEV corpus, the best method being the NUKTI one. Curiously, we did not try to propose any Sure alignment but did receive a credit for it for two of the variants we submitted.

variant	T.	Prec.	Rec.	F-m.	AER
JAPA	P	26.17	74.49	38.73	71.27
JAPA +	S	9.62	67.58	16.84	
NUKTI	P	51.34	53.60	52.44	46.64
NUKTI	S	12.24	86.01	21.43	
	p	63.09	65.87	64.45	30.6

Table 5: Performance of the 3 alignments we submitted for the TEST corpus. $T.$ stands for the type of alignment (Sure or Possible).

6 Discussion

We proposed two methods for aligning an English-Inuktitut bitext at the word level and a combination of both. The best of these methods involves computing an association score between English tokens and Inuktitut substrings. It relies on a greedy algorithm we specifically devised for the task and which seeks a local optimum of a cumulative function of log-likelihood ratio scores. This method obtained a precision and a recall above 63% and 65% respectively.

We believe this method could easily be improved. First, it has some intrinsic limitations, as for instance, the fact that NUKTI only recognizes `1-n` cepts and do not handle at all unaligned words. Indeed, our method is not even suited to aligning English sentences with fewer words than their respective Inuktitut counterpart. Second, the greedy search we devised is fairly aggressive and only explores a tiny bit of the full search. Last, the computation of the association scores is fairly time-consuming.

Our idea of redefining word alignment as a sentence alignment task did not work well; but at the same time, we adapted poorly JAPA to this task. In particular, JAPA does not benefit here from all the potential of the underlying cognate system because of the scarcity of these cognates in very small sequences (words).

If we had to work on this task again, we would consider the use of a morphological analyzer. Unfortunately, it is only after the submission deadline that we learned of the existence of such a tool for Inuktitut[3].

Acknowledgement

We are grateful to Alexandre Patry who turned the JAPA aligner into a nicely written and efficient C++ program.

References

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1).

W. A. Gale and K. W. Church. 1993. A Program for Aligning Sentences in Bilingual Corpora. In *Computational Linguistics*, volume 19, pages 75–102.

P. Langlais, M. Simard, and J. Véronis. 1998. Methods and Practical Issues in Evaluating Alignment Techniques. In *36th annual meeting of the ACL*, Montreal, Canada.

P. Langlais. 1997. A System to Align Complex Bilingual Corpora. QPSR 4, TMH, Stockholm, Sweden.

J. Martin, H. Johnson, B. Farley, and A. Maclachlan. 2003. Aligning and Using an English-Inuktitut Parallel Corpus. In *Building and using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 115–118, Edmonton, Canada.

F.J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:19–51.

M. Simard, G.F. Foster, and P. Isabelle. 1992. Using Cognates to Align Sentences in Bilingual Corpora. In *Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81.

[3]See `http://www.inuktitutcomputing.ca/Uqailaut/`

Models for Inuktitut-English Word Alignment

Charles Schafer and Elliott Franco Drábek
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
{*cschafer,edrabek*}*@cs.jhu.edu*

Abstract

This paper presents a set of techniques for bitext word alignment, optimized for a language pair with the characteristics of Inuktitut-English. The resulting systems exploit cross-lingual affinities at the sublexical level of syllables and substrings, as well as regular patterns of transliteration and the tendency towards monotonicity of alignment. Our most successful systems were based on classifier combination, and we found different combination methods performed best under the target evaluation metrics of F-measure and alignment error rate.

1 Introduction

Conventional word-alignment methods have been successful at treating many language pairs, but may be limited in their ability to generalize beyond the Western European language pairs for which they were originally developed, to pairs which exhibit more complex divergences in word order, morphology and lexical granularity. Our approach to Inuktitut-English alignment was to carefully consider the data in identifying difficulties particular to Inuktitut-English as well as possible simplifying assumptions. We used these observations to construct a novel weighted finite-state transducer alignment model as well as a specialized transliteration model. We combined these customized systems with 3 systems based on IBM Model 4 alignments under several methods of classifier combination. These combination strategies allowed us to produce multiple submissions targeted at the distinct evaluation measures via a precision/recall tradeoff.

2 Special Characteristics of the Inuktitut-English Alignment Problem

Guided by the discussion of Inuktitut in Mallon (1999), we examined the Nunavut Hansards training and hand-labeled trial data sets in order to identify special challenges and exploitable characteristics of the Inuktitut-English word alignment problem. We were able to identify three: (1) Importance of sublexical Inuktitut units; (2) 1-to-N Inuktitut-to-English alignment cardinality; (3) Monotonicity of alignments.

2.1 Types and Tokens

Inuktitut has an extremely productive agglutinative morphology, and an orthographic word may combine very many individual morphemes. As a result, in Inuktitut-English bitext we observe Inuktitut sentences with many fewer word tokens than the corresponding English sentences; the ratio of English to Inuktitut tokens in the training corpus is 1.85.[1] This suggests the importance of looking below the Inuktitut word level when computing lexical translation probabilities (or alignment affinities). To reinforce the point, consider that the ratio of training corpus types to tokens is 0.007 for English, and 0.194 for Inuktitut. In developing a customized word alignment solution for Inuktitut-English, a major goal was to handle the huge number of Inuktitut word types seen only once in the training corpus (337798 compared to 8792 for English), without demanding the development of a morphological analyzer.

2.2 Alignment

Considering English words in English sentence order, 4.7% of their alignments to Inuktitut were found to be *retrograde*; that is, involving a decrease in Inuktitut word position with respect to the previous English word's aligned Inuktitut position. Since this method of counting retrograde alignments would assign a low count to mass movements of large contiguous chunks, we also measured the number of inverted alignments over all pairs of English word positions. That is, the sum

$$\Sigma_e \Sigma_{a=1}^{a=|e|-1} \Sigma_{b=a+1}^{b=|e|} \Sigma_{i_1 \in I(e,a)} \Sigma_{i_2 \in I(e,b)} (1 \text{ if } i_1 > i_2)$$

was computed over all Inuktitut alignment sets $I(e, x)$, for e the English sentence and x the English word position. Dividing this sum by the obvious denominator (replacing $(1 \text{ if } i_1 > i_2)$ with (1) in the sum) yielded a value of 1.6% inverted alignments.

Table 1 shows a histogram of alignment cardinalities for both English and Inuktitut. Ninety-four percent of English word tokens, and ninety-nine percent of those having a non-null alignment, align to exactly one Inuktitut word. In development of a specialized word aligner for this language pair (Section 3), we made use of the observed reliability of these two properties, monotonicity and 1-to-N cardinality.

3 Alignment by Weighted Finite-State Transducer Composition

We designed a specialized alignment system to handle the above-mentioned special characteristics of Inuktitut-

[1] Though this ratio increases to 2.21 when considering only longer sentences (20 or more English words), ignoring common short, formulaic sentence pairs such as (Hudson Bay) | (sanikiluaq) .

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 79–82,
Ann Arbor, June 2005.

	% Words Having Specified Alignment Cardinality							
	NULL	**1**	**2**	**3**	**4**	**5**	**6**	**7**
English	5	94	<1	<1	0	0	0	0
Inuktitut	3	43	20	14	10	5	3	2

Table 1: Alignment cardinalities for English-Inuktitut word alignment, computed over the trial data.

English alignment. Our weighted finite-state transducer (WFST) alignment model, illustrated in Figure 1, structurally enforces monotonicity and 1-to-N cardinality, and exploits sublexical information by incorporating association scores between English words and Inuktitut word substrings, based on co-occurrence in aligned sentences. For each English word, an association score was computed not only with each Inuktitut word, but also with each Inuktitut character string of length ranging from 2 to 10 characters. This is similar to the technique described in Martin et al. (2003) as part of their construction of a bilingual glossary from English-Inuktitut bitext. However, our goal is different and we keep *all* the English-Inuktitut associations, rather than selecting only the "best" ones using a greedy method, as do they. Additionally, before extracting all substrings from each Inuktitut word, we added a special character to the word's beginning and end (e.g., *makkuttut* → *_makkuttut_*), in order to exploit any preferences for word-initial or -final placement.

The heuristic association score chosen was $p(word_e|word_i) \times p(word_i|word_e)$, computed over all the aligned sentence pairs. We have in the past observed this to be a useful indicator of word association, and it has the nice property of being in the range (0,1].

The WFST aligner is a composition of 4 transducers.[2] The structure of the entire WFST composition enforces monotonicity, Inuktitut-to-English 1-N cardinality, and Inuktitut word fertilities ranging between 1 and 7. This model was implemented using the ATT finite-state toolkit (Mohri et al., 1997). In Figure 1, **[1]** is a linear transducer mapping each English position in a particular English test sentence to the word at that position. It is constructed so as to force each English word to participate in exactly 1 alignment. **[2]** is a single-state transducer mapping English word to Inuktitut substrings (or full words) with weights derived from the association scores.[3] **[3]** is a transducer mapping Inuktitut substrings (and full words) to their position in the Inuktitut test sentence. Its construction allows a single Inuktitut position to correspond to multiple English positions, while enforcing monotonicity. **[4]** is a transducer regulating the allowed "fertility" values of Inuktitut words; each Inuktitut word is permitted a fertility of between 1 and 7. The fertility values are assigned the probabilities corresponding to observed relative frequencies in the *trial* data, and

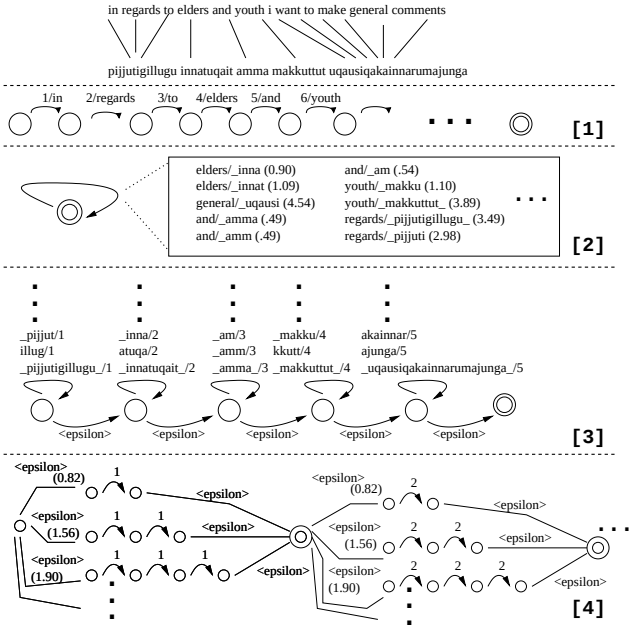


Figure 1: WFST alignment system in composition order, instantiated for an example sentence from the development (trial) data. To save space, only a representative portion of each machine is drawn. Transition weights are costs in the tropical **(min,+)** semiring, derived from negative logs of probabilities and association scores. Nonzero costs are indicated in parentheses.

are not conditioned on the identity of the Inuktitut word.

4 English-Inuktitut Transliteration

Although in this corpus English and Inuktitut are both written in Roman characters, English names are significantly transformed when rendered in Inuktitut text. Consider the following English/Inuktitut pairs from the training corpus: **Chartrand/saaturaan**, **Chretien/kurittian** and the set of training corpus-attested Inuktitut renderings of **Williams**, **Campbell**, and **McLean** shown in Table 2(A) (which does not include variations containing the common **-mut** lexeme, meaning "to [a person]" (Mallon, 1999)).

Clearly, not only does the English-to-Inuktitut transformation radically change the name string, it does so in a nondeterministic way which appears to be influenced not only by the phonological preferences of Inuktitut but also by differing pronunciations of the name in question and possibly by differing conventions of translators (note, for example, **maklain** versus **mikliin** for **McLean**).

We trained a probabilistic finite-state transducer (FST) to identify English-Inuktitut transliterated pairs in aligned sentences. Training string pairs were acquired from the training bitext in the following manner. Whenever single instances of corresponding honorifics were found in a sentence pair – these included the correspondences (Ms , mis); (Mrs , missa/missis); (Mr ,

[2] Bracketed numbers in the following discussion refer to the component transducers as illustrated in Figure 1.

[3] Transducers **[2]** and **[4]** are shared across all sentence decodings.

(A)		(B)			
Williams	**McLean**	**k**		**sh**	
ailiams	makalain	k	-4.2	s	-7.2
uialims	makkalain	q	-6.2		
uilialums	maklaain			**w**	
uiliam	maklain	**b**		ui	-5.8
uiliammas	maklainn	p	-4.3	v	-6.1
uiliams	maklait	v	-5.0		
uilians	makli			**o**	
uliams	maklii	**z**		a	-4.2
viliams	makliik	j	-5.2	aa	-4.6
	makliin	s	-5.8	uu	-4.9
Campbell	maklin			u	-5.1
kaampu	malain	**ch**			
kaampul	matliin	s	-5.6	**u**	
kaamvul	miklain	k	-6.8	uu	-5.5
kamvul	mikliin			u	-5.6
	miklin			a	-6.2

Table 2: (A) Training-corpus-attested renderings of **Williams**, **Campbell**, and **McLean**. (B) Top learned Inuktitut substitutions and their log probabilities for several English *(shown underlined)* orthographic characters (and character sequences). Where top substitutions for English characters are shown, none equal or better were omitted.

mista/mistu) – the immediately following capitalized English words (up to 2) were extracted and the same number of Inuktitut words were extracted to be used as training pairs. Thus, given the appearance in aligned sentences of "Mr. Quirke" and "mista kuak", the training pair (Quirke,kuak) would be extracted. Common distractions such as "Mr Speaker" were filtered out. In order to focus on the native English name problem (Inuktitut name rendering into English is much less noisy) the English extractions were required to have appeared in a large, news-corpus-derived English wordlist. This procedure resulted in a conservative, high-quality list of 434 unique name pairs. The probabilistic FST model we selected was that of a memoryless (single-state) transducer representing a joint distribution over character substitutions, English insertions, and Inuktitut insertions. This model is identical to that presented in Ristad and Yianilos (1997). Prior to training, common English digraphs (e.g., "th" and "sh") were mapped to unique single characters, as were doubled consonants. Inuktitut "ng" and common two-vowel sequences were also mapped to unique single characters to elicit higher-quality results from the memoryless transduction model employed. Some results of the transducer training are displayed in Table 2(B). Probabilistic FST weight training was accomplished using the Dyna modeling language and DynaMITE parameter optimization toolkit (Eisner et al, 2004). The transliteration modeling described here differs from such previous transliteration work as Stalls and Knight (1998) in that there is no explicit modeling of pronunciation, only a direct transduction between written forms.

In applying transliteration on trial/test data, the following criteria were used to select English words for transliteration: *(1) Word is capitalized (2) Word is not in the exclusion list.*[4] For the top-ranked transliteration of the English word present in the Inuktitut sentence, all occurrences of that word in that sentence are marked as aligned to the English word.

We have yet to evaluate English-Inuktitut transliteration in isolation on a large test set. However, accuracy on the workshop trial data was 4/4 hypotheses correct, and on test data 2/6 correct. Of the 4 incorrect test hypotheses, 2 were mistakes in identifying the correct transliteration, and 2 mistakes resulted from attempting to transliterate an English word such as "Councillors" which should not be transliterated. Even with a relatively low accuracy, the transliteration model, which is used only as an individual voter in combination systems, is unlikely to vote for the incorrect choice of another system. Its purpose under system combination is to push a good alignment link hypothesis up to the required vote threshold.[5]

5 IBM Model 4 Alignments

As a baseline and contributor to our combination systems, we ran GIZA++ (Och and Ney, 2000), to produce alignments based on IBM Model 4. The IBM alignment models are asymmetric, requiring that one language be idenitifed as the "e" language, whose words are allowed many links each, and the other as the "f" language, whose words are allowed at most one link each. Although the observed alignment cardinalities naturally suggest identifying Inuktitut as the "e" language and English as the "f" language, we ran both directions for completeness.

As a crude first attempt to capture sublexical correspondences in the absence of a method for morpheme segmentation, we developed a rough syllable segmenter (spending approximately 2 person-hours), ran GIZA++ to produce alignments treating the syllables as words, and chose, for each English word, the Inuktitut word or words the largest number of whose syllables were linked to it.

In the nomenclature of our results tables, **giza++ syllabized** refers to the latter system, **giza++ E(1)-I(N)** represents GIZA++ run with English as the "e" language, and **giza++ E(N)-I(1)** sets English as the "f" language.

6 System Performance and Combination Methods

We observed the 4 main systems (3 GIZA++ variants and WFST) to have significantly different performance profiles in terms of precision and recall. Consistently, WFST

[4] Exclusion list was compiled as follows: (a) capitalized words in 2000 randomly selected English training sentences were examined, Words such as *Clerk*, *Federation*, and *Fisheries*, which are frequently capitalized but should not be transliterated, were put into the exclusion list; in addition, any word with frequency > 50 in the training corpus was excluded, on the rationale that common-enough words would have well-estimated translation probabilities already. 50 may seem like a high threshold until one considers the high variability of the transliteration process as demonstrated in Table 2(A).

[5] Refer to Section 6 for detailed descriptions of voting.

SYSTEM	P	R	F	AER	$\|H\|/\|T\|$
Individual system performance **Trial Data**					
giza++ E(1)-I(N)	63.4	26.6	37.5	32.9	0.42
giza++ E(N)-I(1)	68.2	59.4	63.5	28.6	0.87
giza++ syllabized	83.6	44.5	58.1	**18.3**	0.53
WFST	70.3	72.7	**71.5**	27.8	1.03
Combination system performance **Trial Data**					
F/AER Emphasis	85.4	63.5	72.9	12.3	0.74
AER Emphasis (1)	92.6	44.2	59.9	**8.8**	0.48
AER Emphasis (2)	95.1	38.0	54.3	9.5	0.40
F Emphasis	74.8	77.6	**76.2**	21.9	1.04
Recall Emphasis	66.9	82.1	73.8	28.9	1.23
Individual system performance **Test Data**					
giza++ E(1)-I(N)	49.7	18.6	27.0	45.2	0.37
giza++ E(N)-I(1)	64.6	56.2	60.1	32.7	0.87
giza++ syllabized	84.9	44.0	57.9	**15.6**	0.52
WFST	65.4	68.3	**66.8**	33.7	1.04
(submitted) *Combination system performance* **Test Data**					
F/AER Emphasis	84.4	58.6	69.2	14.3	0.69
AER Emphasis (1)	90.7	39.4	54.9	11.5	0.43
AER Emphasis (2)	96.7	32.3	48.4	**9.5**	0.33
F Emphasis	70.7	73.8	**72.2**	26.7	1.04
Recall Emphasis	62.6	81.7	70.1	34.2	1.31

Table 3: System performance evaluated on trial and test data. The precision, recall and F-measure cited are the unlabeled version ("probable," in the nomenclature of this shared task). The gold standard truth for trial data contained 710 alignments. The test gold standard included 1972 alignments. The column $|H|/|T|$ lists ratio of hypothesis set size to truth set size for each system.

won out on F-measure while **giza++ syllabized** attained better alignment error rate (AER). Refer to Table 3 for details of performance on trial and test data.

We investigated a number of system combination methods, three of which were finally selected for use in submitted systems. There were two basic methods of combination: *per-link voting* and *per-English-word* voting.[6] In per-link voting, an alignment link is included if it is proposed by at least a certain number of the participating individual systems. In per-English-word voting, the best outgoing link is chosen for each English word (the link which is supported by the greatest number of individual systems). Any ties are broken using the WFST system choice. A high-recall variant of per-English-word voting was included in which ties at vote-count 1 (indicating a low-confidence decision) are not broken, but rather all systems' choices are submitted as hypotheses.

The transliteration model described in Section 4 was included as a voter in each combination system, though it made few hypotheses (6 on the test data). Composition of the submitted systems was as follows: **F/AER Emphasis** - per-link voting with decision criterion $>= 2$ votes, over all 5 described systems (WFST, 3 GIZA++ variants, transliteration). **AER Emphasis (I)** per-link voting, $>= 2$ votes, over all systems except giza++ E(N)-I(1). **AER Emphasis (II)** per-link voting, $>= 3$ votes, over all systems. **F Emphasis** per-English-word voting, over all systems, using WFST as tiebreaker. **Recall Emphasis** per-English-word voting, over all systems, high-recall variant.

We elected to submit these systems because each tailors to a distinct evaluation criterion (as suggested by the naming convention). Experiments on trial data convinced us that minimizing AER and maximizing F-measure in a single system would be difficult. Minimizing AER required such high-precision results that the tradeoff in recall greatly lowered F-measure. It is interesting to note that system combination does provide a convenient means for adjusting alignment precision and recall to suit the requirements of the problem or evaluation standard at hand.

[6]Combination methods we elected not to submit included voting with trained weights and various stacked classifiers. The reasoning was that with such a small development data set – 25 sentences – it was unsafe to put faith in any but the simplest of classifier combination schemes.

7 Conclusions

We have presented several individual and combined systems for word alignment of Inuktitut-English bitext. The most successful individual systems were those targeted to the specific characteristics of the language pair. The combined systems generally outperformed the individual systems, and different combination methods were able to optimize for performance under different evaluation metrics. In particular, per-English-word voting performed well on F-measure, while per-link voting performed well on AER.

Acknowledgements: *Many thanks to Eric Goldlust, David Smith, and Noah Smith for help in using the* Dyna *language.*

References

J. Eisner, E. Goldlust, and N. A. Smith. 2004. Dyna: A declarative language for implementing dynamic programs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Companion Volume, pages 218-221.

M. Mallon. 1999. Inuktitut linguistics for technocrats. Technical report, Ittukuluuk Language Programs, Iqaluit, Nunavut, Canada.

J. Martin, H. Johnson, B. Farley, and A. Maclachlan. 2003. Aligning and using an English-Inuktitut parallel corpus. In *Proceedings of Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, HLT-NAACL 2003.

M. Mohri, F. Pereira, and M. Riley. 1997.
ATT General-purpose finite-state machine software tools.
http://www.research.att.com/sw/tools/fsm/.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

E. S. Ristad and P. N. Yianilos. 1997. Learning string edit distance. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 287–295.

B. Stalls and K. Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

Improved HMM Alignment Models for Languages with Scarce Resources

Adam Lopez
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
alopez@cs.umd.edu

Philip Resnik
Institute for Advanced Computer Studies
Department of Linguistics
University of Maryland
College Park, MD 20742
resnik@umiacs.umd.edu

Abstract

We introduce improvements to statistical word alignment based on the Hidden Markov Model. One improvement incorporates syntactic knowledge. Results on the workshop data show that alignment performance exceeds that of a state-of-the art system based on more complex models, resulting in over a 5.5% absolute reduction in error on Romanian-English.

1 Introduction

The most widely used alignment model is IBM Model 4 (Brown et al., 1993). In empirical evaluations it has outperformed the other IBM Models and a Hidden Markov Model (HMM) (Och and Ney, 2003). It was the basis for a system that performed very well in a comparison of several alignment systems (Dejean et al., 2003; Mihalcea and Pedersen, 2003). Implementations are also freely available (Al-Onaizan et al., 1999; Och and Ney, 2003).

The IBM Model 4 search space cannot be efficiently enumerated; therefore it cannot be trained directly using Expectation Maximization (EM). In practice, a sequence of simpler models such as IBM Model 1 and an HMM Model are used to generate initial parameter estimates and to enumerate a partial search space which can be expanded using hill-climbing heuristics. IBM Model 4 parameters are then estimated over this partial search space as an approximation to EM (Brown et al., 1993; Och and Ney, 2003). This approach yields good results, but it has been observed that the IBM Model 4 performance is only slightly better than that of the underlying HMM Model used in this bootstrapping process (Och and Ney, 2003). This is illustrated in Figure 1.

Based on this observation, we hypothesize that implementations of IBM Model 4 derive most of their performance benefits from the underlying HMM Model. Furthermore, owing to the simplicity of HMM Models, we believe that they are more conducive to study and improvement than more complex models such as IBM Model 4. We illustrate this point by introducing modifications to the HMM model which improve performance.

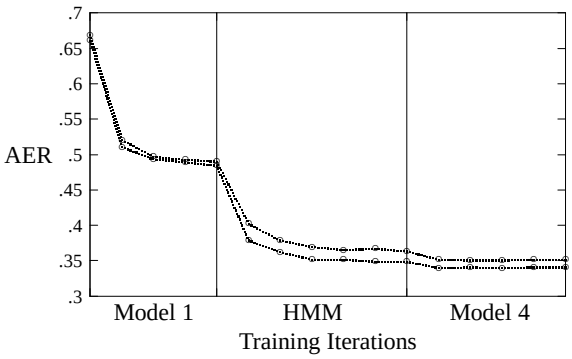


Figure 1: The improvement in Alignment Error Rate (AER) is shown for both $P(\mathbf{f}|\mathbf{e})$ and $P(\mathbf{e}|\mathbf{f})$ alignments on the Romanian-English development set over several iterations of the IBM Model 1 → HMM → IBM Model 4 training sequence.

2 HMMs and Word Alignment

The objective of word alignment is to discover the word-to-word translational correspondences in a bilingual corpus of S sentence pairs, which we denote $\{(\mathbf{f}^{(s)}, \mathbf{e}^{(s)}) : s \in [1,S]\}$. Each sentence pair $(\mathbf{f},\mathbf{e}) = (f_1^M, e_1^N)$ consists of a sentence $\mathbf{f}$ in one language and its translation $\mathbf{e}$ in the other, with lengths M and N, respectively. By convention we refer to $\mathbf{e}$ as the English sentence and $\mathbf{f}$ as the French sentence. Correspondences in a sentence are represented by a set of links between words. A link (f_j, e_i) denotes a correspondence between the ith word e_i of $\mathbf{e}$ and the jth word f_j of $\mathbf{f}$.

Many alignment models arise from the conditional distribution $P(\mathbf{f}|\mathbf{e})$. We can decompose this by introducing the hidden alignment variable $\mathbf{a} = a_1^M$. Each element of $\mathbf{a}$ takes on a value in the range $[1,N]$. The value of a_i determines a link between the ith French word f_i and the a_ith English word e_{a_i}. This representation introduces

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 83–86,
Ann Arbor, June 2005.

an asymmetry into the model because it constrains each French word to correspond to exactly one English word, while each English word is permitted to correspond to an arbitrary number of French words. Although the resulting set of links may still be relatively accurate, we can symmetrize by combining it with the set produced by applying the complementary model $P(\mathbf{e}|\mathbf{f})$ to the same data (Och and Ney, 2000b). Making a few independence assumptions we arrive at the decomposition in Equation 1. [1]

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) = \prod_{i=1}^{M} d(a_i|a_{i-1}) \cdot t(f_i|e_{a_i}) \quad (1)$$

We refer to $d(a_i|a_{i-1})$ as the *distortion model* and $t(f_i|e_{a_i})$ as the *translation model*. Conveniently, Equation 1 is in the form of an HMM, so we can apply standard algorithms for HMM parameter estimation and maximization. This approach was proposed in Vogel et al. (1996) and subsequently improved (Och and Ney, 2000a; Toutanova et al., 2002).

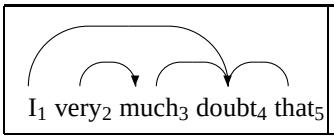
2.1 The Tree Distortion Model

Equation 1 is adequate in practice, but we can improve it. Numerous parameterizations have been proposed for the distortion model. In our *surface distortion* model, it depends only on the distance $a_i - a_{i-1}$ and an automatically determined word class $C(e_{a_{i-1}})$ as shown in Equation 2. It is similar to (Och and Ney, 2000a). The word class $C(e_{a_{i-1}})$ is assigned using an unsupervised approach (Och, 1999).

$$d(a_i|a_{i-1}) = p(a_i|a_i - a_{i-1}, C(e_{a_{i-1}})) \quad (2)$$

The surface distortion model can capture local movement but it cannot capture movement of structures or the behavior of long-distance dependencies across translations. The intuitive appeal of capturing richer information has inspired numerous alignment models (Wu, 1995; Yamada and Knight, 2001; Cherry and Lin, 2003). However, we would like to retain the simplicity and good performance of the HMM Model.

We introduce a distortion model which depends on the *tree distance* $\tau(e_i, e_k) = (w, x, y)$ between each pair of English words e_i and e_k. Given a dependency parse of e_1^M, w and x represent the respective number of dependency links separating e_i and e_k from their closest common ancestor node in the parse tree. [2] The final element $y = \{1$ if $i > k$; 0 otherwise$\}$ is simply a binary indicator of the linear relationship of the words within the surface string. Tree distance is illustrated in Figure 2.



$\tau(\text{I}_1, \text{very}_2) = (1,2,0)$
$\tau(\text{very}_2, \text{I}_1) = (2,1,1)$
$\tau(\text{I}_1, \text{doubt}_4) = (1,0,0)$
$\tau(\text{that}_5, \text{I}_1) = (1,1,1)$

Figure 2: Example of tree distances in a sentence from the Romanian-English development set.

In our *tree distortion* model, we condition on the tree distance and the part of speech $T(e_{i-1})$, giving us Equation 3.

$$d(a_i|a_{i-1}) = p(a_i, |\tau(e_{a_i}, e_{a_{i-1}}), T(e_{a_{i-1}})) \quad (3)$$

Since both the surface distortion and tree distortion models represent $p(a_i|a_{i-1})$, we can combine them using linear interpolation as in Equation 4.

$$\begin{aligned} &d(a_i|a_{i-1}) = \\ &\quad \lambda_{C(e_{a_{i-1}}),T(e_{a_{i-1}})} p(a_i|\tau(e_{a_i}, e_{a_{i-1}}), T(e_{a_{i-1}})) + \\ &\quad (1 - \lambda_{C(e_{a_{i-1}}),T(e_{a_{i-1}})}) p(a_i|a_i - a_{i-1}, C(e_{a_{i-1}})) \end{aligned} \quad (4)$$

The $\lambda_{C,T}$ parameters can be initialized from a uniform distribution and trained with the other parameters using EM. In principle, any number of alternative distortion models could be combined with this framework.

2.2 Improving Initialization

Our HMM produces reasonable results if we draw our initial parameter estimates from a uniform distribution. However, we can do better. We estimate the initial translation probability $t(f_j|e_i)$ from the smoothed log-likelihood ratio $LLR(e_i, f_j)^{\phi_1}$ computed over sentence cooccurrences. Since this method works well, we apply $LLR(e_i, f_j)$ in a single reestimation step shown in Equation 5.

$$t(f|e) = \frac{LLR(f|e)^{\phi_2} + n}{\sum_{e'} LLR(f|e')^{\phi_2} + n \cdot |V|} \quad (5)$$

In reestimation $LLR(f|e)$ is computed from the expected counts of f and e produced by the EM algorithm. This is similar to Moore (2004); as in that work, $|V| = 100,000$, and ϕ_1, ϕ_2, and n are estimated on development data.

We can also use an improved initial estimate for distortion. Consider a simple distortion model $p(a_i|a_i - a_{i-1})$. We expect this distribution to have a maximum near $P(a_i|0)$ because we know that words tend to retain their locality across translation. Rather than wait for this to occur, we use an initial estimate for the distortion model given in Equation 6.

[1] We ignore the sentence length probability $p(M|N)$, which is not relevant to word alignment. We also omit discussion of HMM start and stop probabilities, and normalization of $t(f_i|e_{a_i})$, although we find in practice that attention to these details can be beneficial.

[2] The tree distance could easily be adapted to work with phrase-structure parses or tree-adjoining parses instead of dependency parses.

corpus	n	ϕ_1	ϕ_2	α	symmetrization	n^{-1}	ϕ_1^{-1}	ϕ_2^{-1}	α^{-1}
English-Inuktitut	1^{-4}	1.0	1.75	-1.5	$\cap$	5^{-4}	1.0	1.75	-1.5
Romanian-English	5^{-4}	1.5	1.0	-2.5	refined (Och and Ney, 2000b)	5^{-4}	1.5	1.0	-2.5
English-Hindi	1^{-4}	1.5	3.0	-2.5	$\cup$	1^{-2}	1.0	1.0	-1.0

Table 1: Training parameters for the workshop data (see Section 2.2). Parameters n, ϕ_1, ϕ_2, and α were used in the initialization of $P(\mathbf{f}|\mathbf{e})$ model, while n^{-1}, ϕ_1^{-1}, ϕ_2^{-1}, and α^{-1} were used in the initialization of the $P(\mathbf{e}|\mathbf{f})$ model.

corpus	type	**HMM limited (Eq. 2)**			**HMM unlimited (Eq. 4)**			IBM Model 4		
		P	R	AER	P	R	AER	P	R	AER
English-Inuktitut	P(f|e)	.4962	.6894	.4513	–	–	–	.4211	.6519	.5162
	P(e|f)	.5789	**.8635**	.3856	–	–	–	.5971	.8089	.3749
	$\cap$	**.8916**	.6280	**.2251**	–	–	–	.8682	.5700	.2801
English-Hindi	P(f|e)	.5079	.4769	.5081	.5057	.4748	.5102	.5219	.4223	.5332
	P(e|f)	.5566	.4429	.5067	.5566	.4429	.5067	.5652	.3939	.5358
	$\cup$	.4408	**.5649**	.5084	.4365	.5614	.5088	**.4543**	.5401	**.5065**
Romanian-English	P(f|e)	.6876	**.6233**	.3461	.6876	**.6233**	.3461	.6828	.5414	.3961
	P(e|f)	.7168	.6217	.3341	.7155	.6205	.3354	.7520	.5496	.3649
	refined	.7377	.6169	**.3281**	.7241	.6215	.3311	**.7620**	.5134	.3865

Table 2: Results on the workshop data. The systems highlighted in bold are the ones that were used in the shared task. For each corpus, the last row shown represents the results that were actually submitted. Note that for English-Hindi, our self-reported results in the unlimited task are slightly lower than the original results submitted for the workshop, which contained an error.

$$d(a_i|a_{i-1}) = \begin{cases} |a_i - a_{i-1}|^{\alpha}/Z, \alpha < 0 & \text{if } a_i \neq a_{i-1}. \\ 1/Z & \text{if } a_i = a_{i-1}. \end{cases} \quad (6)$$

We choose Z to normalize the distribution. We must optimize α on a development set. This distribution has a maximum when $|a_i - a_{i-1}| \in \{-1, 0, 1\}$. Although we could reasonably choose any of these three values as the maximum for the initial estimate, we found in development that the maximum of the surface distortion distribution varied with $C(e_{a_i-1})$, although it was always in the range $[-1, 2]$.

2.3 Does NULL Matter in Asymmetric Alignment?

Och and Ney (2000a) introduce a NULL-alignment capability to the HMM alignment model. This allows any word f_j to link to a special NULL word – by convention denoted e_0 – instead of one of the words e_1^N. A link (f_j, e_0) indicates that f_j does not correspond to any word in **e**. This improved alignment performance in the absence of symmetrization, presumably because it allows the model to be conservative when evidence for an alignment is lacking.

We hypothesize that NULL alignment is unnecessary for asymmetric alignment models when we symmetrize using intersection-based methods (Och and Ney, 2000b). The intuition is simple: if we don't permit NULL alignments, then we expect to produce a high-recall, low-precision alignment; the intersection of two such alignments should mainly improve precision, resulting in a high-recall, high-precision alignment. If we allow NULL alignments, we may be able produce a high-precision, low-recall asymmetric alignment, but symmetrization by intersection will not improve recall.

3 Results with the Workshop Data

In our experiments, the dependency parse and parts of speech are produced by minipar (Lin, 1998). This parser has been used in a much different alignment model (Cherry and Lin, 2003). Since we only had parses for English, we did not use tree distortion in the application of $P(\mathbf{e}|\mathbf{f})$, needed for symmetrization.

The parameter settings that we used in aligning the workshop data are presented in Table 1. Although our prior work with English and French indicated that intersection was the best method for symmetrization, we found in development that this varied depending on the characteristics of the corpus and the type of annotation (in particular, whether the annotation set included probable alignments). The results are summarized in Table 2. It shows results with our HMM model using both Equations 2 and 4 as our distortion model, which represent

the unlimited and limited resource tracks, respectively. It also includes a comparison with IBM Model 4, for which we use a training sequence of IBM Model 1 (5 iterations), HMM (6 iterations), and IBM Model 4 (5 iterations). This sequence performed well in an evaluation of the IBM Models (Och and Ney, 2003).

For comparative purposes, we show results of applying both $P(\mathbf{f}|\mathbf{e})$ and $P(\mathbf{e}|\mathbf{f})$ prior to symmetrization, along with results of symmetrization. Comparison of the asymmetric and symmetric results largely supports the hypothesis presented in Section 2.3, as our system generally produces much better recall than IBM Model 4, while offering a competitive precision. Our symmetrized results usually produced higher recall and precision, and lower alignment error rate.

We found that the largest gain in performance came from the improved initialization. The combined distortion model (Equation 4), which provided a small benefit over the surface distortion model (Equation 2) on the development set, performed slightly worse on the test set.

We found that the dependencies on $C(e_{a_{i-1}})$ and $T(e_{a_{i-1}})$ were harmful to the $P(\mathbf{f}|\mathbf{e})$ alignment for Inuktitut, and did not submit results for the unlimited resources configuration. However, we found that alignment was generally difficult for all models on this particular task, perhaps due to the agglutinative nature of Inuktitut.

4 Conclusions

We have proposed improvements to the largely overlooked HMM word alignment model. Our improvements yield good results on the workshop data. We have additionally shown that syntactic information can be incorporated into such a model; although the results are not superior, they are competitive with surface distortion. In future work we expect to explore additional parameterizations of the HMM model, and to perform extrinsic evaluations of the resulting alignments by using them in the parameter estimation of a phrase-based translation model.

Acknowledgements

This research was supported in part by ONR MURI Contract FCPO.810548265. The authors would like to thank Bill Byrne, David Chiang, Okan Kolak, and the anonymous reviewers for their helpful comments.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation: Final report. In *Johns Hopkins University 1999 Summer Workshop on Language Engineering*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, Jun.

Colin Cherry and Dekang Lin. 2003. A probability model to improve word alignment. In *ACL Proceedings*, Jul.

Herve Dejean, Eric Gaussier, Cyril Goutte, and Kenji Yamada. 2003. Reducing parameter space for word alignment. In *Proceedings of the Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 23–26, May.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems*, May.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10, May.

Robert C. Moore. 2004. Improving IBM word-alignment model 1. In *ACL Proceedings*, pages 519–526, Jul.

Franz Josef Och and Hermann Ney. 2000a. A comparison of alignment models for statistical machine translation. In *COLING Proceedings*, pages 1086–1090, Jul.

Franz Josef Och and Hermann Ney. 2000b. Improved statistical alignment models. In *ACL Proceedings*, pages 440–447, Oct.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison on various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *EACL Proceedings*, pages 71–76, Jun.

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. 2002. Extensions to hmm-based statistical word alignment models. In *EMNLP*, pages 87–94, Jul.

Stephan Vogel, Hermann Ney, and Christoph Tillman. 1996. Hmm-based word alignment in statistical machine translation. In *COLING Proceedings*, pages 836–841, Aug.

Dekai Wu. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1328–1335, Aug.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *ACL Proceedings*.

Symmetric Probabilistic Alignment

Ralf D. Brown **Jae Dong Kim** **Peter J. Jansen** **Jaime G. Carbonell**

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{ralf,jdkim,pjj,jgc}@cs.cmu.edu

Abstract

We recently decided to develop a new alignment algorithm for the purpose of improving our Example-Based Machine Translation (EBMT) system's performance, since subsentential alignment is critical in locating the correct translation for a matched fragment of the input. Unlike most algorithms in the literature, this new Symmetric Probabilistic Alignment (SPA) algorithm treats the source and target languages in a symmetric fashion.

In this short paper, we outline our basic algorithm and some extensions for using context and positional information, and compare its alignment accuracy on the Romanian-English data for the shared task with IBM Model 4 and the reported results from the prior workshop.

1 Symmetric Probabilistic Alignment (SPA)

In subsentential alignment, mappings are produced from words or phrases in the source language sentence and those words or phrases in the target language sentence that best express their meaning.

An alignment algorithm takes as input a bilingual corpus consisting of corresponding sentence pairs and strives to find the best possible alignment in the second for selected n-grams (sequences of n words) in the first language. The alignments are based on a number of factors, including a bilingual dictionary (preferably a probabilistic one), the position of the words, invariants such as numbers and punctuation, and so forth.

For our baseline algorithm, we make the following simplifying assumptions, each of which we intend to relax in future work, and the last of which has already been partially relaxed:

1. A fixed bilingual probabilistic dictionary is available.
2. Fragments (word sequences) are translated independently of surrounding context.
3. Contiguous fragments of source language text are translated into contiguous fragments in the target language text.

Unlike the work of (Marcu and Wong, 2002), our alignment algorithm is not generative and does not use the idea of a bag of concepts from which the phrases in the sentence pair arise. It is, rather, intended to find the corresponding target-language phrase given a specific source-language phrase of interest, as required by our EBMT system after finding a match between the input and the training data (Brown, 2004).

1.1 Baseline Algorithm

Our baseline algorithm is based on maximizing the probability of bi-directional translations of individual words between a selected n-gram in the source language and every possible n-gram in the corresponding paired target language sentence. No positional preference assumptions are made, nor are any length preservation assumptions made. That is, an n-gram may translate to an m-gram, for any values of n or m bounded by the source and target sentence lengths, respectively. Finally a smoothing factor is used to avoid singularities (i.e. avoiding zero-probabilities for unknown words, or words never translated before in a way consistent with the dictionary).

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 87–90,
Ann Arbor, June 2005.

Given a source-language sentence

$$S1 : s_0, s_1, ..., s_i, ..., s_{i+k}, ..., s_n \quad (1)$$

in the bilingual corpus, where $s_i, ..., s_{i+k}$ is a phrase of interest, and the corresponding target language sentence S2 is

$$S2 : t_0, t_1, ..., t_j, ..., t_{j+l}, ..., t_m \quad (2)$$

the values of j and l are to be determined.

Then the segment we try to obtain is the target fragment $\hat{F}_T$ with the highest probability of all possible fragments of S2 to be a mutual translation with the given source fragment, or

$$\hat{F}_T = \text{argmax}_{\{F_T\}} \left(p(s_i, ..., s_{i+k} \leftrightarrow t_j, ..., t_{j+l})\right) \quad (3)$$

All possible segments can be checked in $O(m^2)$ time, where m is the target language length, because we will check m 1-word segments, $m-1$ two-word segments, and so on. If we bound the target language n-grams to a maximal length k, then the complexity is linear, i.e. $O(km)$.

The score of the best possible alignment is computed as follows: Let L_T be the Target Language Vocabulary, s a source word, t_i be target segment words, and $V = \{t_i \in \{L_T\} | i \geq 1\}$ the translation word set of s,

We define the *translation relation probability* $p(Tr(s) \in \{t_0, t_1, ..., t_k\})$ as follows:

1. $p(Tr(s) \in \{t_0, t_1, ..., t_k\}) = \max(p(t_i|s))$ for all $t_i \in \{t_0, t_1, ..., t_k\}$ when $\{t_i | t_i \in \{t_0, t_1, ..., t_k\}\}$ is not empty.
2. $p(Tr(s) \in \{t_0, t_1, ..., t_k\}) = 0$ otherwise.

Then the score of the best alignment is

$$S_{\hat{F}_T} = \max_{\{F_T\}} S_{F_T} \quad (4)$$

where the score can be written as two components

$$S_{F_T} = P_1 \times P_2 \quad (5)$$

which can be further specified as

$$P_1 = \left(\prod_{m=0}^{k} \max \left(p\left(Tr(s_{i+m}) \in \{t_{j...j+l}\}\right), \epsilon\right) \right)^{\frac{1}{k+1}} \quad (6)$$

$$P_2 = \left(\prod_{n=0}^{l} \max \left(p\left(Tr(t_{j+n}) \in \{s_{i...i+k}\}\right), \epsilon\right) \right)^{\frac{1}{l+1}} \quad (7)$$

where ϵ is a very small probability used as a *smoothing value*.

1.2 Length Penalty

The ratio between source and target segment (n-gram) lengths should be comparable to the ratio between the lengths of the source and target sentences, though certainly variation is possible. Therefore, we add a penalty function to the alignment probability that increases with the discrepancy between the two ratios.

Let the length of the source language segment be i and the length of a target language segment under consideration be j. Given a source language sentence length of n (in the corpus sentence containing the fragment) and its corresponding target language length of m. The *expected target segment length* is then given by $\hat{j} = i \times \frac{m}{n}$. Further defining an *allowable difference* AD, our implementation calculates the length penalty LP as follows, with the value of the exponent determined empirically:

$$LP_{F_T} = \min \left(\left(\frac{|j - \hat{j}|}{AD} \right)^4, 1 \right) \quad (8)$$

The score for a segment including the penalty function is then:

$$S_{F_T} \leftarrow S_{F_T} \times (1 - LP_{F_T}) \quad (9)$$

Note that, as intended, the score is forced to 0 when the length difference $|j - \hat{j}| > AD$.

1.3 Distortion Penalty

For closely-related language pairs which tend to have similar word orders, we introduce a distortion penalty to penalize the alignment score of any candidate target fragment which is out of the expected position range. First, we calculate C_E, the expected center of the candidate target fragment using C_{F_S}, the center of the source fragment and the ratio of target- to source-sentence length.

$$C_E = C_{F_S} * \frac{m}{n} \quad (10)$$

Then we calculate an allowed distance limit of the center $D_{allowed}$ using a constant distance limit value DL and the ratio of actual target sentence length to average target sentence length.

$$D_{allowed} = DL * \frac{m}{m_{average}} \quad (11)$$

Let D_{actual} be the actual distance difference between the candidate target fragment's center and the expected center, and set

$$S_{F_T} \leftarrow \begin{cases} 0, \text{if} D_{actual} \geq D_{allowed} \\ \frac{S_{F_T}}{(D_{actual} - D_{allowed} + 1)^2}, \text{otherwise} \end{cases} \quad (12)$$

Furthermore, we think that we can apply this penalty to language pairs which have lower word-order similarities than e.g. French-English. Because there might exist certain positional relationships between such language pairs, if we can calculate the expected position using each language's sentence structure, we can apply a distortion penalty to the candidate alignments.

1.4 Anchor Context

If the adjacent words of the source fragment and the candidate target fragment are translations of each other, we expect that this alignment is more likely to be correct. We boost S_{F_T} with the anchor context alignment score S_{AC_p},

$$S_{AC_p} = P(s_{i-1} \leftrightarrow t_{j-1}) * P(s_{i+k} \leftrightarrow t_{j+l}) \quad (13)$$

$$S_{F_T} \leftarrow (S_{F_T})^{\lambda} * (S_{AC_p})^{1-\lambda} \quad (14)$$

Empirically, we found this combination gives the best score for French-English when $\lambda = 0.6$ and for Romanian-English when $\lambda = 0.8$, and leads to better results than the similar formula

$$S_{F_T} \leftarrow \lambda * S_{F_T} + (1 - \lambda) * S_{AC_p} \quad (15)$$

2 Experimental Design

In previous work (Kim et al., 2005), we tested our alignment method on a set of French-English sentence pairs taken from the Canadian Hansard corpus and on a set of English-Chinese sentence pairs, and compared the results to human alignments. For the present workshop, we chose to use the Romanian-English data which had been made available.

Due to a lack of time prior to the period of the shared task, we merely re-used the parameters which had been tuned for French-English, rather than tuning the alignment parameters specifically for the development data.

SPA was run under three experimental conditions. In the first, labeled "SPA (c)" in Tables 1 and 2, SPA was instructed to examine only contiguous target phrases as potential alignments for a given source phrase. In the second, labeled "SPA (n)", a noncontiguous target alignment consisting of two contiguous segments with a gap between them was permitted in addition to contiguous target alignments. The third condition ("SPA (h)") examined the impact of a small amount of manual alignment information on the selection of contiguous alignments. Unlike the first two conditions, the presence of additional data beyond the training corpus forces SPA(h) into the Unlimited Resources track.

We had a native Romanian speaker hand-align 204 sentence pairs from the training corpus, and extracted 732 distinct translation pairs from those alignments, of which 450 were already present in the automatically-generated dictionaries. The new translation pairs were added to the dictionaries for the SPA(h) condition and the translation probabilities for the existing pairs were increased to reflect the increased confidence in their correctness. Had more time been available, we would have investigated more sophisticated means of integrating the human knowledge into the translation dictionaries.

3 Results and Conclusions

Table 1 compares the performance of SPA on what is now the development data against the submissions with the best AER values reported by (Mihalcea and Pedersen, 2003) for the participants in the 2003 workshop, including CMU, MITRE, RALI, University of Alberta, and XRCE [1]. As SPA generates only SURE alignments, the values in Table 1 are SURE alignments under the NO-NULL-Align scoring condition for all systems except Fourday, which did not generate SURE alignments.

Despite the fact that SPA was designed specifically for phrase-to-phrase alignments rather than the

[1] Citations for individual participants' papers have been omitted for space reasons; all appear in the same proceedings.

Method	Prec%	Rec%	F1%	AER
SPA (c)	64.47	62.68	63.56	36.44
SPA (n)	64.38	62.70	63.53	36.47
SPA (h)	64.61	62.55	63.56	36.44
Fourday	52.83	42.86	47.33	52.67
UMD.RE.2	58.29	49.99	53.82	46.61
BiBr	70.65	55.75	62.32	41.39
Ralign	92.00	45.06	60.49	35.24
XRCEnolm	82.65	62.44	71.14	28.86

Table 1: Romanian-English alignment results (Development Set, NO-NULL-Align)

word-to-word alignments needed for the shared task and was not tuned for this corpus, its performance is competitive with the best of the systems previously used for the shared task. We thus decided to submit runs for the official 2005 evaluation, whose resulting scores are shown in Table 2.

On the development set, noncontiguous alignments resulted in slightly lower precision than contiguous alignments, which was not unexpected, but recall does not increase enough to improve F1 or AER. The modified dictionaries improved precision slightly, as anticipated, but lowered recall sufficiently to have no net effect on F1 or AER.

The evaluation set proved to be very similar in difficulty to the development data, resulting in scores that were very close to those achieved on the dev-test set. Noncontiguous alignments again proved to have a very small negative effect on AER resulting from reduced precision, but this time the altered dictionaries for SPA(h) resulted in a substantial reduction in recall, considerably harming overall performance.

After the shared task was complete, we performed some tuning of the alignment parameters for the Romanian-English development test set, and found that the French-English-tuned parameters were close to optimal in performance. The AER on the development test set for the SPA(c) contiguous alignments condition decreased from 36.44% to 36.11% after the re-tuning.

4 Future Work

Enhancements in the extraction of word-to-word alignments from what is fundamentally a phrase-to-phrase alignment algorithm could probably further improve results on the Romanian-English data. We also intend to investigate principled, seamless integration of manual alignments and dictionaries with probabilistic ones, since the *ad hoc* method proved detrimental. Finally, a more detailed performance analysis is in order, to determine whether the close balance of precision and recall is inherent in the bidirectionality of the algorithm or merely coincidence.

Method	Prec%	Recall%	F1%	AER%
SPA (c)	64.96	61.34	63.10	36.90
SPA (n)	64.91	61.34	63.07	36.93
SPA (h)	64.60	60.54	62.50	37.50

Table 2: Evaluation results (NO-NULL-Align)

5 Acknowledgements

We would like to thank Lucian Vlad Lita for providing manual alignments.

References

Ralf D. Brown. 2004. A Modifi ed Burrows-Wheeler Transform for Highly-Scalable Example-Based Translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004)*, volume 3265 of *Lecture Notes in Artificial Intelligence*, pages 27–36. Springer Verlag, September-October. `http://www.cs.cmu.edu/~ralf/papers.html`.

Jae Dong Kim, Ralf D. Brown, Peter J. Jansen, and Jaime G. Carbonell. 2005. Symmetric Probabilistic Alignment for Example-Based Translation. In *Proceedings of the Tenth Workshop of the European Association for Machine Translation (EAMT-05)*, May. (to appear).

Daniel Marcu and William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, July. `http://www.isi.edu/~marcu/papers.html`.

Rada Mihalcea and Ted Pedersen. 2003. An Evaluation Exercise for Word Alignment. In *Proceedings of the HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10. Association for Computational Linguistics, May.

ISI's Participation in the Romanian-English Alignment Task

Alexander Fraser
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6601
fraser@isi.edu

Daniel Marcu
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6601
marcu@isi.edu

Abstract

We discuss results on the shared task of Romanian-English word alignment. The baseline technique is that of symmetrizing two word alignments automatically generated using IBM Model 4. A simple vocabulary reduction technique results in an improvement in performance. We also report on a new alignment model and a new training algorithm based on alternating maximization of likelihood with minimization of error rate.

1 Introduction

ISI participated in the WPT05 Romanian-English word alignment task. The system used for baseline experiments is two runs of IBM Model 4 (Brown et al., 1993) in the GIZA++ (Och and Ney, 2003) implementation, which includes smoothing extensions to Model 4. For symmetrization, we found that Och and Ney's "refined" technique described in (Och and Ney, 2003) produced the best AER for this data set under all experimental conditions.

We experimented with a statistical model for inducing a stemmer cross-lingually, but found that the best performance was obtained by simply lowercasing both the English and Romanian text and removing all but the first four characters of each word.

We also tried a new model and a new training criterion based on alternating the maximization of likelihood and minimization of the alignment error rate. For these experiments, we have implemented an alignment package for IBM Model 4 using a hill-climbing search and Viterbi training as described in (Brown et al., 1993), and extended this to use new submodels. The starting point is the final alignment generated using GIZA++'s implementation of IBM Model 1 and the Aachen HMM model (Vogel et al., 1996).

Paper organization: Section 2 is on the baseline, Section 3 discusses vocabulary reduction, Section 4 introduces our new model and training method, Section 5 describes experiments, Section 6 concludes.

We use the following notation: e refers to an English sentence composed of English words labeled e_i. f refers to a Romanian sentence composed of Romanian words labeled f_j. a is an alignment of e to f. We use the term "Viterbi alignment" to denote the most probable alignment we can find, rather than the true Viterbi alignment.

2 Baseline

To train our systems, Model 4 was trained two times, first using Romanian as the source language and then using English as the source language. For each training, we ran 5 iterations of Model 1, 5 iterations of the HMM model and 3 iterations of Model 4. For the distortion calculations of Model 4, we removed the dependencies on Romanian and English word classes. We applied the "union", "intersection" and "refined" symmetrization metrics (Och and Ney, 2003) to the final alignments output from training, as well as evaluating the two final alignments directly.

We tried to have a strong baseline. GIZA++ has many free parameters which can not be estimated using Maximum Likelihood training. We did not use

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 91–94,
Ann Arbor, June 2005.

the defaults, but instead used settings which produce good AER results on French/English bitext. We also optimized $p0$ on the 2003 test set (using AER), rather than using likelihood training. Turning off the extensions to GIZA++ and training $p0$ as in (Brown et al., 1993) produces a substantial increase in AER.

3 Vocabulary Size Reduction

Romanian is a Romance language which has a system of suffixes for inflection which is richer than English. Given the small amount of training data, we decided that vocabulary size reduction was desirable. As a baseline for vocabulary reduction, we tried reducing words to prefixes of varying sizes for both English and Romanian after lowercasing the corpora. We also tried Porter stemming (Porter, 1997) for English.

(Rogati et al., 2003) extended Model 1 with an additional hidden variable to represent the split points in Arabic between the prefix, the stem and the suffix to generate a stemming for use in Cross-Lingual Information Retrieval. As in (Rogati et al., 2003), we can find the most probable stemming given the model, apply this stemming, and retrain our word alignment system. However, we can also use the modified model directly to find the best word alignment without converting the text to its stemmed form.

We introduce a variable r_j for the Romanian stem and a variable s_j for the Romanian suffix (which when concatenated together give us the Romanian word f_j) into the formula for the probability of generating a Romanian word f_j using an alignment a_j given only an English sentence e. We use the index z to denote a particular stemming possibility. For a given Romanian word the stemming possibilities are simply every possible split point where the stem is at least one character (this includes the null suffix).

$$p(f_j, a_j|e) = \sum_z p(r_{j,z}, s_{j,z}, a_j|e) \quad (1)$$

If the assumption is made that the stem and the suffix are generated independently from e, we can assume conditional independence.

$$p(f_j, a_j|e) = \sum_z p(r_{j,z}, a_j|e)p(s_{j,z}, a_j|e) \quad (2)$$

We performed two sets of experiments, one set where the English was stemmed using the Porter stemmer and one set where each English word was stemmed down to its first four characters. We tried the best performing scoring heuristic for Arabic from (Rogati et al., 2003) where $p(s_{j,z}, a_j|e)$ is modeled using the heuristic $p(s_{j,z}|l_j)$ where $s_{j,z}$ is the Romanian suffix, and l_j is the last letter of the Romanian word f_j; these adjustments are updated during EM training. We also tried several other approximations of $p(s_{j,z}, a_j|e)$ with and without updates in EM training. We were unable to produce better results and elected to use the baseline vocabulary reduction technique for the shared task.

4 New Model and Training Algorithm

Our motivation for a new model and a new training approach which combines likelihood maximization with error rate minimization is threefold:

- Maximum Likelihood training of Model 4 is not sufficient to find good alignments
- We would like to model factors not captured by IBM Model 4
- Using labeled data could help us produce better alignments, but we have very few labels

We create a new model and train it using an algorithm which has a step which increases likelihood (like one iteration in the EM algorithm), alternating with a step which decreases error. We accomplish this by:

- grouping the parameters of Model 4 into 5 submodels
- implementing 6 new submodels
- combining these into a single log-linear model with 11 weights, λ_1 to λ_{11}, which we group into the vector λ
- defining a search algorithm for finding the alignment of highest probability given the submodels and λ
- devising a method for finding a λ which minimizes alignment error given fixed submodels and a set of gold standard alignments
- inventing a training method for alternating steps which estimate the submodels by increasing likelihood with steps which set λ to decrease alignment error

The submodels in our new alignment model are listed in table 1, where for ease of exposition we

Table 1: Submodels used for alignment

1	$t(f_j\|e_i)$	TRANSLATION PROBABILITIES
2	$n(\phi_i\|e_i)$	FERTILITY PROBABILITIES, ϕ_i IS THE NUMBER OF WORDS GENERATED BY THE ENGLISH WORD e_i
3	$null$	PARAMETERS USED IN GENERATING ROMANIAN WORDS FROM ENGLISH NULL WORD (INCLUDING $p0$, $p1$)
4	$d_1(\triangle j)$	MOVEMENT (DISTORTION) PROBABILITIES OF FIRST ROMANIAN WORD GENERATED FROM ENGLISH WORD
5	$d_{>1}(\triangle j)$	MOVEMENT (DISTORTION) PROBABILITIES OF OTHER ROMANIAN WORDS GENERATED FROM ENGLISH WORD
6		TTABLE ESTIMATED FROM INTERSECTION OF TWO STARTING ALIGNMENTS FOR THIS ITERATION
7		TRANSLATION TABLE FROM ENGLISH TO ROMANIAN MODEL 1 ITERATION 5
8		TRANSLATION TABLE FROM ROMANIAN TO ENGLISH MODEL 1 ITERATION 5
9		BACKOFF FERTILITY (FERTILITY ESTIMATED OVER ALL ENGLISH WORDS)
10		ZERO FERTILITY ENGLISH WORD PENALTY
11		NON-ZERO FERTILITY ENGLISH WORD PENALTY

consider English to be the source language and Romanian the target language.

The log-linear alignment model is specified by equation 3. The model assigns non-zero probabilities only to 1-to-many alignments, like Model 4. (Cettolo and Federico, 2004) used a log-linear model trained using error minimization for the translation task, 3 of the submodels were taken from Model 4 in a similar way to our first 5 submodels.

$$p_\lambda(a, f|e) = \frac{exp(\sum_m \lambda_m h_m(f, a, e))}{\sum_{f,e,a} exp(\sum_m \lambda_m h_m(f, a, e))} \quad (3)$$

Given λ, the alignment search problem is to find the alignment a of highest probability according to equation 3. We solve this using the local search defined in (Brown et al., 1993).

We set λ as follows. Given a sequence A of alignments we can calculate an error function, $E(A)$. For these experiments average sentence AER was used. We wish to minimize this error function, so we select λ accordingly:

$$\underset{\lambda}{\operatorname{argmin}} \sum_{\tilde{a}} E(\tilde{a}) \delta(\tilde{a}, (\underset{a}{\operatorname{argmax}}\, p_\lambda(a, f|e))) \quad (4)$$

Maximizing performance for all of the weights at once is not computationally tractable, but (Och, 2003) has described an efficient one-dimensional search for a similar problem. We search over each λ_m (holding the others constant) using this technique to find the best λ_m to update and the best value to update it to. We repeat the process until no further gain can be found.

Our new training method is:

REPEAT

- Start with submodels and lambda from previous iteration
- Find Viterbi alignments on entire training corpus using new model (similar to E-step of Model 4 training)
- Reestimate submodel parameters from Viterbi alignments (similar to M-step of Model 4 Viterbi training)
- Find a setting for λ that reduces AER on discriminative training set (new D-step)

We use the first 148 sentences of the 2003 test set for the discriminative training set. 10 settings for λ are found, the hypothesis list is augmented using the results of 10 searches using these settings, and then another 10 settings for λ are found. We then select the best λ. The discriminative training regimen is otherwise similar to (Och, 2003).

5 Experiments

Table 2 provides a comparison of our baseline systems using the "refined" symmetrization metric with the best limited resources track system from WPT03 (Dejean et al., 2003) on the 2003 test set. The best results are obtained by stemming both English and Romanian words to the first four letters, as described in section 2.

Table 3 provides details on our shared task submission. RUN1 is the word-based baseline system. RUN2 is the stem-based baseline system. RUN4 uses only the first 6 submodels, while RUN5 uses all 11 submodels. RUN3 had errors in processing, so we omit it.

Results:

- Our new 1-to-many alignment model and training method are successful, producing decreases of 0.03 AER when the source is Romanian, and 0.01 AER when the source is English.

Table 2: Summary of results for 2003 test set

SYSTEM	STEM SIZES	AER
XEROX "NOLEM-ER-56K"		0.289
BASELINE	NO PROCESSING	0.284
BASELINE	ENG PORTER / ROM 4	0.251
BASELINE	ENG 4 / ROM 4	0.248

Table 3: Full results on shared task submissions (blind test 2005)

RUN NAMES	STEM SIZES	SOURCE ROM	SOURCE ENG	UNION	INTERSECTION	REFINED
ISI.RUN1	NO PROCESSING	0.3834	0.3589	0.3590	0.3891	0.3165
ISI.RUN2	ENG 4 / ROM 4	0.3056	0.2880	0.2912	0.3041	0.2675
ISI.RUN4	ENG 4 / ROM 4	0.2798	0.2833	0.2773	0.2862	0.2663
ISI.RUN5	ENG 4 / ROM 4	0.2761	0.2778	0.2736	0.2807	0.2655

- These decreases do not translate to a large improvement in the end-to-end task of producing many-to-many alignments with a balanced precision and recall. We had a very small decrease of 0.002 AER using the "refined" heuristic.
- The many-to-many alignments produced using "union" and the 1-to-1 alignments produced using "intersection" were also improved.
- It may be a problem that we trained $p0$ using likelihood (it is in submodel 3) rather than optimizing $p0$ discriminatively as we did for the baseline.

6 Conclusion

- Considering multiple stemming possibilities for each word seems important.
- Alternating between increasing likelihood and decreasing error rate is a useful training approach which can be used for many problems.
- Our model and training method improve upon a strong baseline for producing 1-to-many alignments.
- Our model and training method can be used with the "intersection" heuristic to produce higher quality 1-to-1 alignments
- Models which can directly model many-to-many alignments and do not require heuristic symmetrization are needed to produce higher quality many-to-many alignments. Our training method can be used to train them.

7 Acknowledgments

This work was supported by DARPA-ITO grant NN66001-00-1-9814 and NSF grant IIS-0326276.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Mauro Cettolo and Marcello Federico. 2004. Minimum error training of log-linear translation models. In *Proc. of the International Workshop on Spoken Language Translation*, pages 103–106, Kyoto, Japan.

Herve Dejean, Eric Gaussier, Cyril Goutte, and Kenji Yamada. 2003. Reducing parameter space for word alignment. In *HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Alberta, July.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

M. F. Porter. 1997. An algorithm for suffi x stripping. In *Readings in information retrieval*, pages 313–316, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Monica Rogati, Scott McCarley, and Yiming Yang. 2003. Unsupervised learning of arabic stemming using a parallel corpus. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, August.

Experiments Using MAR for Aligning Corpora*

Juan Miguel Vilar
Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Castellón (Spain)
jvilar@lsi.uji.es

Abstract

We present some experiments conducted within the context of one of the shared tasks of the ACL 2005 Workshop on Building and Using Parallel Texts. We have employed a new model for finding the alignments. This new model takes a recursive approach in order to find the alignments. As its computational costs are quite high, a method for splitting the training sentences in smaller parts is used.

1 Introduction

We present the experiments we conducted within the context of the shared task of the track on building and using parallel texts for languages with scarce resources of the ACL 2005 Workshop on Building and Using Parallel Texts. The aim of the task was to align the words of sentence pairs in different language pairs. We have participated using the Romanian-English corpora.

We have used a new model, the MAR (from the Spanish initials of Recursive Alignment Model) that allowed us to find structured alignments that were later transformed in a more conventional format. The basic idea of the model is that the translation of a sentence can be obtained in three steps: first, the sentence is divided in two parts; second, each part is translated separately using the same process; and third, the two translations are joined. The high computational costs associated with the training of the model made it necessary to split the training pairs in smaller parts using a simple heuristic.

Initial work with this model can be seen in (Vilar Torres, 1998). A detailed presentation can be found in (Vilar and Vidal, 2005). This model shares some similarities with the stochastic inversion transduction grammars (SITG) presented by Wu in (Wu, 1997). The main point in common is the number of possible alignments between the two models. On the other hand, the parametrizations of SITGs and the MAR are completely different. The generative process of SITGs produces simultaneously the input and output sentences and the parameters of the model refer to the rules of the nonterminals. This gives a clear symmetry to both input and output sentences. Our model clearly distinguishes an input and output sentence and the parameters are based on observable properties of the sentences (their lengths and the words composing them). Also, the idea of splitting the sentences until a simple structure is found in the Divisive Clustering presented in (Deng et al., 2004). Again, the main difference is in the probabilistic modeling of the alignments. In Divisive Clustering a uniform distribution on the alignments is assumed while MAR uses a explicit parametrization.

The rest of the paper is structured as follows: the next section gives an overview of the MAR, then we explain the task and how the corpora were split, after that, how the alignments were obtained is explained, finally the results and conclusions are presented.

*Work partially supported by Bancaixa through the project "Sistemas Inductivos, Estadísticos y Estructurales, para la Traducción Automática (SIEsTA)".

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 95–98,
Ann Arbor, June 2005.

2 The MAR

We provide here a brief description of the model, a more detailed presentation can be found in (Vilar and Vidal, 2005). The idea is that the translation of a sentence $\bar{x}$ into a sentence $\bar{y}$ can be performed in the following steps[1]:

(a) If $\bar{x}$ is small enough, IBM's model 1 (Brown et al., 1993) is employed for the translation.

(b) If not, a cut point is selected in $\bar{x}$ yielding two parts that are independently translated applying the same procedure recursively.

(c) The two translations are concatenated either in the same order that they were produced or second first.

2.1 Model parameters

Apart from the parameters of model 1 (a stochastic dictionary and a discrete distribution of lenghts), each of the steps above defines a set of parameters. We will consider now each set in turn.

Deciding the submodel The first decision is whether to use IBM's model 1 or to apply the MAR recursively. This decision is taken on account of the length of $\bar{x}$. A table is used so that:

$$\Pr(\text{IBM} \mid \bar{x}) \approx \mathcal{M}_I(|\bar{x}|),$$
$$\Pr(\text{MAR} \mid \bar{x}) \approx \mathcal{M}_M(|\bar{x}|).$$

Clearly, for every $\bar{x}$ we have that $\Pr(\text{IBM} \mid \bar{x}) + \Pr(\text{MAR} \mid \bar{x}) = 1$.

Deciding the cut point It is assumed that the probability of cutting the input sentence at a given position b is most influenced by the words around it: x_b and x_{b+1}. We use a table $\mathcal{B}$ such that:

$$\Pr(b \mid \bar{x}) \approx \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})}.$$

That is, a weight is assigned to each pair of words and they are normalized in order to obtaing a proper probability distribution.

[1]We use the following notational conventions. A string or sequence of words is indicated by a bar like in $\bar{x}$, individual words from the sequence carry a subindex and no bar like in x_i, substrings are indicated with the first and last position like in $\bar{x}_i^j$. Finally, when the final position of the substring is also the last of the string, a dot is used like in $\bar{x}_i^{\cdot}$

Deciding the concatenation direction The direction of the concatenation is also decided as a function of the two words adjacent to the cut point, that is:

$$\Pr(D \mid b, \bar{x}) \approx \mathcal{D}_D(x_b, x_{b+1}),$$
$$\Pr(I \mid b, \bar{x}) \approx \mathcal{D}_I(x_b, x_{b+1}),$$

where D stands for *direct* concatenation (i.e. the translation of $\bar{x}_1^b$ will precede the translation of $\bar{x}_{b+1}^{\cdot}$) and I stands for *inverse*. Clearly, $\mathcal{D}_D(x_b, x_{b+1}) + \mathcal{D}_I(x_b, x_{b+1}) = 1$ for every pair (x_b, x_{b+1}).

2.2 Final form of the model

With these parameters, the final model is:

$$\begin{aligned} p_T(\bar{y} \mid \bar{x}) = & \\ & \mathcal{M}_I(|\bar{x}|) p_I(\bar{y} \mid \bar{x}) \\ & + \mathcal{M}_M(|\bar{x}|) \sum_{b=1}^{|\bar{x}|-1} \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})} \\ & \cdot \Bigg(\mathcal{D}_D(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_1^c \mid \bar{x}_1^b) p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_{b+1}^{\cdot}) \\ & + \mathcal{D}_I(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_1^b) p_T(\bar{y}_1^c \mid \bar{x}_{b+1}^{\cdot}) \Bigg) \end{aligned}$$

were p_I represents the probability assigned by model 1 to a pair of sentences.

2.3 Model training

The training of the model parameters is done maximizing the likelihood of the training sample. For each training pair $(\bar{x}, \bar{y})$ and each parameter P relevant to it, the value of

$$\mathcal{C}(P) = \frac{P}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial\, P} \qquad (1)$$

is computed. This corresponds to the *counts* of P in that pair. As the model is polynomial on all its parameters except for the cuts (the $\mathcal{B}$'s), Baum-Eagon's inequality (Baum and Eagon, 1967) guarantees that normalization of the counts increases the likelihood of the sample. For the cuts, Gopalakrishnan's inequality (Gopalakrishnan et al., 1991) is used.

Table 1: Statistics of the training corpus. Vocabulary refers to the number of different words.

Language	Sentences	Words	Vocabulary
Romanian	48 481	976 429	48 503
English	48 481	1 029 507	27 053

The initial values for the dictionary are trained using model 1 training and then a series of iterations are made updating the values of every parameter. Some additional considerations are taken into account for efficiency reasons, see (Vilar and Vidal, 2005) for details.

A potential problem here is the large number of parameters associated with cuts and directions: two for each possible pair of words. But, as we are interested only in aligning the corpus, no provision is made for the data sparseness problem.

3 The task

The aim of the task was to align a set of 200 translation pairs between Romanian and English. As training material, the text of 1984, the Romanian Constitution and a collection of texts from the Web were provided. Some details about this corpus can be seen in Table 1.

4 Splitting the corpus

To reduce the high computational costs of training of the parameters of MAR, a heuristic was employed in order to split long sentences into smaller parts with a length less than l words.

Suppose we are to split sentences $\bar{x}$ and $\bar{y}$. We begin by aligning each word in $\bar{y}$ to a word in $\bar{x}$. Then, a score and a translation is assigned to each substring $\bar{x}_i^j$ with a length below l. The translation is produced by looking for the substring of $\bar{y}$ which has a length below l and which has the largest number of words aligned to positions between i and j. The pair so obtained is given a score equal to sum of: (a) the square of the length of $\bar{x}_i^j$; (b) the square of the number of words in the output aligned to the input; and (c) minus ten times the sum of the square of the number of words aligned to a nonempty position out of $\bar{x}_i^j$ and the number of words outside the segment chosen that are aligned to $\bar{x}_i^j$.

These scores are chosen with the aim of reducing the number of segments and making them as "complete" as possible, ie, the words they cover are aligned to as many words as possible.

After the segments of $\bar{x}$ are so scored, the partition of $\bar{x}$ that maximizes the sum of scores is computed by dynamic programming.

The training material was split in parts up to ten words in length. For this, an alignment was obtained by training an IBM model 4 using GIZA++ (Och and Ney, 2003). The test pairs were split in parts up to twenty words. After the split, there were $141\,945$ training pairs and 337 test pairs. Information was stored about the partition in order to be able to recover the correct alignments later.

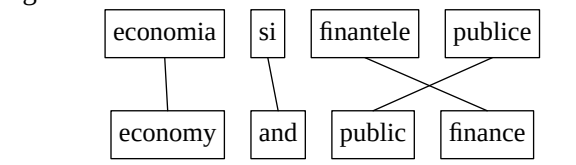
5 Aligning the corpus

The parameters of the MAR were trained as explained above: first ten IBM model 1 iterations were used for giving initial values to the dictionary probabilities and then ten more iterations for retraining the dictionary together with the rest of the parameters.

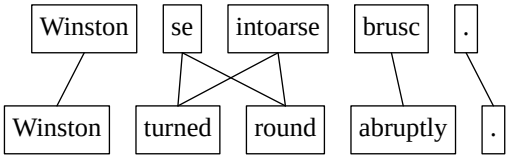
The alignment of a sentence pair has the form of a tree similar to those in Figure 1. Each interior node has two children corresponding to the translation of the two parts in which the input sentence is divided. The leaves of the tree correspond to those segments that were translated by model 1.

As the reference alignments do not have this kind of structure it is necessary to "flatten" them. The procedure we have employed is very simple: if we are in a leaf, every output word is aligned to every input word; if we are in an interior node, the "flat" alignments for the children are built and then combined. Note that the way leaves are labeled tends to favor recall over precision.

The flat alignment corresponding to the trees of Figure 1 are:



and



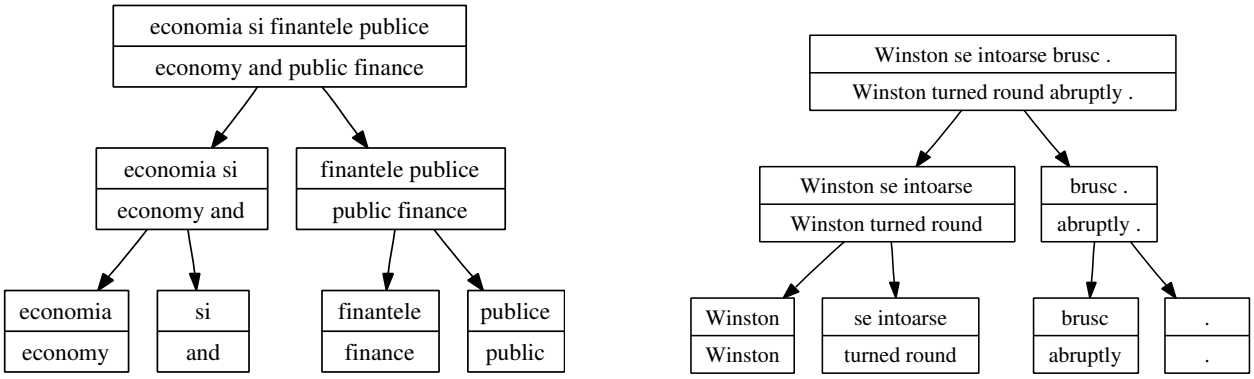


Figure 1: Two trees representing the alignment of two pair of sentences.

Precision	Recall	F-Measure	AER
0.5404	0.6465	0.5887	0.4113

Table 2: Results for the task

6 Results and discussion

The results for the alignment can be seen in Table 2. As mentioned above, there is a certain preference for recall over precision. For comparison, using GIZA++ on the split corpus yields a precision of 0.6834 and a recall of 0.5601 for a total AER of 0.3844.

Note that although the definition of the task allowed to mark the alignment as either *probable* or *sure*, we marked all the alignments as *sure*, so precision and recall measures are given only for sure alignments.

There are aspects that deserve further experimentation. The first is the split of the original corpus. It would be important to evaluate its influence, and to try to find methods of using MAR without any split at all. A second aspect of great importance is the method used for "flattening". The way leaves of the tree are treated probably could be improved if the dictionary probabilities were somehow taken into account.

7 Conclusions

We have presented the experiments done using a new translation model for finding word alignments in parallel corpora. Also, a method for splitting the input before training the models has been presented.

References

Leonard E. Baum and J. A. Eagon. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Yonggang Deng, Shankar Kumar, and William Byrne. 2004. Bitext chunk alignment for statistical machine translation. Research Note 50, CLSP Johns Hopkins University, April.

P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. 1991. An inequality for rational functions with applications to some statistical problems. *IEEE Transactions on Information Theory*, 37(1):107–113, January.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Juan Miguel Vilar and Enrique Vidal. 2005. A recursive statistical translation model. In *Workshop on Building and Using Parallel Texts*, Ann-Arbour (Michigan), June.

Juan Miguel Vilar Torres. 1998. *Aprendizaje de Traductores Subsecuenciales para su empleo en tareas de dominio restringido*. Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia (Spain). (in Spanish).

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Comparison, Selection and Use of Sentence Alignment Algorithms for New Language Pairs

Anil Kumar Singh
LTRC, IIIT
Gachibowli, Hyderabad
India - 500019
anil@research.iiit.net

Samar Husain
LTRC, IIIT
Gachibowli, Hyderabad
India - 500019
samar@iiit.net

Abstract

Several algorithms are available for sentence alignment, but there is a lack of systematic evaluation and comparison of these algorithms under different conditions. In most cases, the factors which can significantly affect the performance of a sentence alignment algorithm have not been considered while evaluating. We have used a method for evaluation that can give a better estimate about a sentence alignment algorithm's performance, so that the best one can be selected. We have compared four approaches using this method. These have mostly been tried on European language pairs. We have evaluated manually-checked and validated English-Hindi aligned parallel corpora under different conditions. We also suggest some guidelines on actual alignment.

1 Introduction

Aligned parallel corpora are collections of pairs of sentences where one sentence is a translation of the other. Sentence alignment means identifying which sentence in the target language (TL) is a translation of which one in the source language (SL). Such corpora are useful for statistical NLP, algorithms based on unsupervised learning, automatic creation of resources, and many other applications.

Over the last fifteen years, several algorithms have been proposed for sentence alignment. Their performance as reported is excellent (in most cases not less than 95%, and usually 98 to 99% and above). The evaluation is performed in terms of precision, and sometimes also recall. The figures are given for one or (less frequently) more corpus sizes. While this does give an indication of the performance of an algorithm, the variation in performance under varying conditions has not been considered in most cases. Very little information is given about the conditions under which evaluation was performed. This gives the impression that the algorithm will perform with the reported precision and recall under all conditions.

We have tested several algorithms under different conditions and our results show that the performance of a sentence alignment algorithm varies significantly, depending on the conditions of testing. Based on these results, we propose a method of evaluation that will give a better estimate of the performance of a sentence alignment algorithm and will allow a more meaningful comparison. Our view is that unless this is done, it will not be possible to pick up the best algorithm for certain set of conditions. Those who want to align parallel corpora may end up picking up a less suitable algorithm for their purposes. We have used the proposed method for comparing four algorithms under different conditions. Finally, we also suggest some guidelines for using these algorithms for actual alignment.

2 Sentence Alignment Methods

Sentence alignment approaches can be categorized as based on sentence length, word correspondence, and composite (where more than one approaches are combined), though other techniques, such as cog-

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 99–106,
Ann Arbor, June 2005.

nate matching (Simard et al., 1992) were also tried. Word correspondence was used by Kay (Kay, 1991; Kay and Roscheisen, 1993). It was based on the idea that words which are translations of each other will have similar distributions in the SL and TL texts. Sentence length methods were based on the intuition that the length of a translated sentence is likely to be similar to that of the source sentence. Brown, Lai and Mercer (Brown et al., 1991) used word count as the sentence length, whereas Gale and Church (Gale and Church, 1991) used character count. Brown, Lai and Mercer assumed prior alignment of paragraphs. Gale and Church relied on some previously aligned sentences as 'anchors'. Wu (Wu, 1994) also used lexical cues from corpus-specific bilingual lexicon for better alignment.

Word correspondence was further developed in IBM Model-1 (Brown et al., 1993) for statistical machine translation. Melamed (Melamed, 1996) also used word correspondence in a different (geometric correspondence) way for sentence alignment. Simard and Plamondon (Simard and Plamondon, 1998) used a composite method in which the first pass does alignment at the level of characters as in (Church, 1993) (itself based on cognate matching) and the second pass uses IBM Model-1, following Chen (Chen, 1993). The method used by Moore (Moore, 2002) also had two passes, the first one being based on sentence length (word count) and the second on IBM Model-1. Composite methods are used so that different approaches can compliment each other.

3 Factors in Performance

As stated above, the performance of a sentence alignment algorithm depends on some identifiable factors. We can even make predictions about whether the performance will increase or decrease. However, as the results given later show, the algorithms don't always behave in a predictable way. For example, one of the algorithms did worse rather than better on an 'easier' corpus. This variation in performance is quite significant and it cannot be ignored for actual alignment (table-1). Some of these factors have been indicated in earlier papers, but these were not taken into account while evaluating, nor were their effects studied.

Translation of a text can be fairly literal or it can be a recreation, with a whole range between these two extremes. Paragraphs and/or sentences can be dropped or added. In actual corpora, there can even be noise (sentences which are not translations at all and may not even be part of the actual text). This can happen due to fact that the texts have been extracted from some other format such as web pages. While translating, sentences can also be merged or split. Thus, the SL and TL corpora may differ in size.

All these factors affect the performance of an algorithm in terms of, say, precision, recall and F-measure. For example, we can expect the performance to worsen if there is an increase in additions, deletions, or noise. And if the texts were translated fairly literally, statistical algorithms are likely to perform better. However, our results show that this does not happen for all the algorithms.

The linguistic distance between SL and TL can also play a role in performance. The simplest measure of this distance is in terms of the distance on the family tree model. Other measures could be the number of cognate words or some measure based on syntactic features. For our purposes, it may not be necessary to have a quantitative measure of linguistic distance. The important point is that for languages that are distant, some algorithms may not perform too well, if they rely on some closeness between languages. For example, an algorithm based on cognates is likely to work better for English-French or English-German than for English-Hindi, because there are fewer cognates for English-Hindi. It won't be without a basis to say that Hindi is more distant from English than is German. English and German belong to the Indo-Germanic branch whereas Hindi belongs to the Indo-Aryan branch. There are many more cognates between English and German than between English and Hindi. Similarly, as compared to French, Hindi is also distant from English in terms of morphology. The *vibhaktis* of Hindi can adversely affect the performance of sentence length (especially word count) as well as word correspondence based algorithms. From the syntactic point of view, Hindi is a comparatively free word order language, but with a preference for the SOV (subject-object-verb) order, whereas English is more of a fixed word order and SVO type language. For sentence length and IBM model-1 based sentence

alignment, this doesn't matter since they don't take the word order into account. However, Melamed's algorithm (Melamed, 1996), though it allows 'non-monotonic chains' (thus taking care of some difference in word order), is somewhat sensitive to the word order. As Melamed states, how it will fare with languages with more word variation than English and French is an open question.

Another aspect of the performance which may not seem important from NLP-research point of view, is its speed. Someone who has to use these algorithms for actual alignment of large corpora (say, more than 1000 sentences) will have to realize the importance of speed. Any algorithm which does worse than O(n) is bound to create problems for large sizes. Obviously, an algorithm that can align 5000 sentences in 1 hour is preferable to the one which takes three days, even if the latter is marginally more accurate. Similarly, the one which takes 2 minutes for 100 sentences, but 16 minutes for 200 sentences will be difficult to use for practical purposes. Actual corpora may be as large as a million sentences. As an estimate of the speed, we also give the runtimes for the various runs of all the four algorithms tested.

Some algorithms, like those based on cognate matching, may even be sensitive to the encoding or notation used for the text. One of the algorithms tested (Melamed, 1996) gave worse performance when we used a notation called ITRANS for the Hindi text, instead of the WX-notation.[1]

4 Evaluation in Previous Work

There have been attempts to systematically evaluate and compare word alignment algorithms (Och and Ney, 2003) but, surprisingly, there has been a lack of such evaluation for sentence alignment algorithms. One obvious problem is the lack of manually aligned and checked parallel corpora.

Two cases where a systematic evaluation was performed are the ARCADE project (Langlais et al., 1996) and Simard et al. (Simard et al., 1992). In the ARCADE project, six alignment systems were evaluated on several different text types. Simard et al. performed an evaluation on several corpus types and corpus sizes. They, also compared the performance of several (till then known) algorithms.

In most of the other cases, evaluation was performed on only one corpus type and one corpus size. In some cases, certain other factors were considered, but not very systematically. In other words, there wasn't an attempt to study the effect of various factors described earlier on the performance. In some cases, the size used for testing was too small. One other detail is that size was sometimes mentioned in terms of number of words, not number of sentences.

5 Evaluation Measures

We have used local (for each run) as well as global (over all the runs) measures of performance of an algorithm. These measures are:

- Precision (local and global)
- Recall (local and global)
- F-measure (local and global)
- 95% Confidence interval of F-measure (global)
- Runtime (local)

6 An Evaluation Scheme

Unless sentence alignment is correct, everything else that uses aligned parallel corpora, such as word alignment (for automatically creating bilingual dictionaries) or statistical machine translation will be less reliable. Therefore, it is important that the best algorithm is selected for sentence alignment. This requires that there should be a way to systematically evaluate and compare sentence alignment algorithms.

To take into account the above mentioned factors, we used an evaluation scheme which can give an estimate of the performance under different conditions. Under this scheme, we calculate the measures given in the previous section along the following dimensions:

- Corpus type
- Corpus size
- Difference in sizes of SL and TL corpora
- Noise

[1] In this notation, capitalization roughly means aspiration for consonants and longer length for vowels. In addition, 'w' represents 't' as in French *entre* and 'x' means something similar to 'd' in French *de*, hence the name of the notation.

We are also considering the corpus size as a factor in performance because the second pass in Moore's algorithm is based on IBM Model-1, which needs training. This training is provided at runtime by using the tentative alignments obtained from the first pass (a kind of unsupervised learning). This means that larger corpus sizes (enough training data) are likely to make word correspondence more effective. Even for sentence length methods, corpus size may play a role because they are based on the distribution of the length variable. The distribution assumption (whether Gaussian or Poisson) is likely to be more valid for larger corpus sizes.

The following algorithms/approaches were evaluated:

- **Brn**: Brown's sentence length (word count) based method, but with Poisson distribution
- **GC**: Church and Gale's sentence length (character count) based method, but with Poisson distribution
- **Mmd**: Melamed's geometric correspondence based method
- **Mre**: Moore's two-pass method (word count plus word correspondence)

For **Brn** and **GC** we used our own implementations. For **Mmd** we used the GMA alignment tool and for **Mre** we used Moore's implementation. Only 1-to-1 mappings were extracted from the output for calculating precision, recall and F-measure, since the test sets had only 1-to-1 alignments. English and Hindi stop lists and a bilingual lexicon were also supplied to the GMA tool. The parameter settings for this tool were kept the same as for English-Malay. For **Brn** and **GC**, the search method was based on the one used by Moore, i.e., searching within a growing diagonal band. Using this search method meant that no prior segmentation of the corpora was needed (Moore, 2002), either in terms of aligned paragraphs (Gale and Church, 1991), or some aligned sentences as anchors (Brown et al., 1991).

We would have liked to study the effect of linguistic distance more systematically, but we couldn't get equivalent manually-checked aligned parallel corpora for other pairs of languages. We have to rely on the reported results for other language pairs, but those results, as mentioned before, do not mention the conditions of testing which we are considering for our evaluation and, therefore, cannot be directly compared to our results for English-Hindi. Still, we did an experiment on the English-French test data (447 sentences) for the shared task in NAACL 2003 workshop on parallel texts (see table-1).

For all our experiments, the text in Hindi was in WX-notation.

In the following sub-sections we describe the details of the data sets that were prepared to study the variation in performance due to various factors.

6.1 Corpus Type

Three different types of corpora were used for the same language pair (English-Hindi) and size. These were EMILLE, ERDC and India Today. We took 2500 sentences from each of these, as this was the size of the smallest corpus.

6.1.1 EMILLE

EMILLE corpus was constructed by the EMILLE project (Enabling Minority Language Engineering), Lancaster University, UK, and the Central Institute of Indian Languages (CIIL), Mysore, India. It consists of monolingual, parallel and annotated corpora for fourteen South Asian languages. The parallel corpus part has a text (200000 words) in English and its translations in Hindi, Bengali, Punjabi, Gujarati and Urdu. The text is from many different domains like education, legal, health, social, and consumer markets. The documents are mostly in simple, formal language. The translations are quite literal and, therefore, we expected this corpus to be the 'easiest'.

6.1.2 ERDC

The ERDC corpus was prepared by Electronic Research and Development Centre, NOIDA, India. It also has text in different domains but it is an unaligned parallel corpus. A project is going on to prepare an aligned and manually checked version of this corpus. We have used a part of it that has already been aligned and manually checked. It was our opinion that the translations in this corpus are less literal and should be more difficult for sentence alignment than EMILLE. We used this corpus for studying the effect of corpus size, in addition to corpus type.

Table 1: Results for Various Corpus Types (Corpus Size = 2500)

Type		Clean, Same Size				Noisy, Same Size				Noisy, Different Size			
		Brn	**GC**	**Mmd**	**Mre**	**Brn**	**GC**	**Mmd**	**Mre**	**Brn**	**GC**	**Mmd**	**Mre**
EMILLE	P	99.3	99.1	85.0	66.8	85.5	87.4	38.2	66.2	87.2	86.5	48.0	65.5
	R	96.0	93.0	80.0	63.2	80.4	80.0	36.2	58.0	81.2	79.1	46.5	57.4
	F	97.6	96.0	82.0	64.9	82.8	83.5	37.2	61.8	84.0	82.6	47.3	61.2
	T	23	23	261	45	47	44	363	64	25	25	413	47
ERDC	P	99.6	99.5	94.2	100.0	85.4	84.4	48.0	96.5	84.6	85.5	50.9	97.7
	R	99.0	99.1	92.7	97.0	81.7	80.6	46.7	78.9	80.5	81.3	49.8	79.1
	F	99.3	99.3	93.4	98.4	83.5	82.4	47.3	86.8	82.5	83.3	50.3	87.1
	T	31	29	1024	85	92	90	2268	124	55	52	3172	101
India Today	P	91.8	93.9	76.4	99.5	71.5	76.7	49.7	94.4	73.6	75.5	51.7	93.4
	R	81.0	83.0	70.6	81.5	61.0	65.5	47.6	67.5	62.4	64.4	50.1	62.6
	F	86.1	88.1	73.4	89.6	65.8	70.7	48.6	78.7	67.6	69.5	50.9	75.0
	T	32	32	755	91	96	101	2120	159	60	68	987	134
English-French	P	100.0	100.0	100.0	100.0	87.4	87.5	77.2	95.2	91.2	93.3	77.7	96.6
	R	100.0	99.3	100.0	99.3	85.5	84.3	81.7	84.6	83.2	83.7	82.6	83.0

P: Precision, *R:* Recall, *F:* F-Measure, *T:* Runtime (seconds)

6.1.3 India Today

India Today is a magazine published in both English and Hindi. We used some parallel text collected from the Internet versions of this magazine. It consists of news reports or articles which appeared in both languages. We expected this corpus to be the most difficult because the translations are often more like adaptations. They may even be rewritings of the English reports or articles in Hindi. This corpus had 2500 sentences.

6.2 Corpus Size

To study the effect of corpus size, the sizes used were 500, 1000, 5000 and 10000. All these data sets were from ERDC corpus (which was expected to be neither very easy nor very difficult).

6.3 Noise and Difference in Sizes of SL and TL Corpora

To see the effect of noise and the difference in sizes of SL and TL corpora, we took three cases for each of the corpus types and sizes:

- Same size without noise
- Same size with noise
- Different size with noise

Three different data sets were prepared for each corpus type and for each corpus size. To obtain such data sets from the aligned, manually checked and validated corpora, we added noise to the corpora. The noise was in the form of sentences from some other unrelated corpus. The number of such sentences was 10% each of the corpus size in the second case and 5% to SL and 15% to the TL in the third case. The sentences were added at random positions in the SL and TL corpora and these positions were recorded so that we could automatically calculate precision, recall and F-measure even for data sets with noise, as we did for other data sets. Thus, each algorithm was tested on (3+4)(3) = 21 data sets.

7 A Limitation

One limitation of our work is that we are considering only 1-to-1 alignments. This is partly due to practical constraints, but also because 1-to-1 alignments are the ones that can be most easily and directly used for linguistic analysis as well as machine learning.

Since we had to prepare a large number of data sets of sizes up to 10000 sentences, manual checking was a major constraint. We had four options. The first was to take a raw unaligned corpus and manually align it. This option would have allowed consideration of 1-to-many, many-to-1, or partial

Table 2: Results for Various Corpus Sizes

Size		Clean, Same Size				Noisy, Same Size				Noisy, Different Size			
		Brn	**GC**	**Mmd**	**Mre**	**Brn**	**GC**	**Mmd**	**Mre**	**Brn**	**GC**	**Mmd**	**Mre**
500	P	99.2	99.2	93.9	99.8	75.4	78.2	57.4	94.3	83.5	87.2	45.4	92.4
	R	98.8	98.8	91.8	95.0	71.0	73.4	56.8	70.0	77.0	80.8	44.8	70.8
	F	99.0	99.0	92.8	97.3	73.1	75.7	57.1	80.4	80.1	83.9	45.1	80.2
	T	9	9	126	14	10	10	148	13	10	10	181	14
1000	P	99.3	99.6	96.4	100.0	84.6	84.6	67.8	96.8	82.2	84.0	47.3	95.1
	R	98.9	99.4	95.1	96.3	81.4	82.2	68.4	73.7	76.3	78.7	46.1	72.7
	F	99.1	99.5	95.7	98.1	83.0	83.4	68.1	83.7	79.1	81.2	46.7	82.4
	T	13	13	278	29	24	23	335	34	15	15	453	30
5000	P	99.8	99.8	93.2	99.9	88.5	88.6	56.1	98.5	85.9	86.6	57.6	97.8
	R	99.4	99.5	91.6	98.2	83.2	83.3	54.9	86.0	81.7	81.3	56.7	86.3
	F	99.6	99.7	92.4	99.1	85.7	85.9	55.4	91.8	83.7	83.9	57.2	91.7
	T	54	53	3481	186	199	185	5248	274	185	174	3639	275
10000	P	99.8	99.9	93.2	100.0	88.0	88.9	59.6	98.5	86.8	88.7	57.2	98.4
	R	99.4	99.6	91.4	98.6	82.9	83.7	58.9	89.9	81.3	82.8	56.2	89.2
	F	99.6	99.7	92.3	99.3	85.4	86.2	59.2	94.0	84.0	85.6	56.6	94.0
	T	102	96	4356	305	370	346	4477	467	345	322	4351	479

alignments. The second option was to pass the text through an alignment tool and then manually check the output for all kinds of alignment. The third option was to check only for 1-to-1 alignments from this output. The fourth option was to evaluate on much smaller sizes.

In terms of time and effort required, there is an order of difference between the first and the second and also between the second and the third option. It is much easier to manually check the output of an aligner for 1-to-1 alignments than to align a corpus from the scratch. We couldn't afford to use the first two options. The fourth option was affordable, but we decided to opt for a more thorough evaluation of 1-to-1 alignments, than for evaluation of all kinds of alignments for smaller sizes. Thus, our starting data sets had only 1-to-1 alignments.

In future, we might extend the evaluation to all kinds of alignments, since the manual alignment currently being done on ERDC corpus includes partial and 1-to-2 or 2-to-1 alignments. Incidentally, there are rarely any 2-to-1 alignments in English-Hindi corpus since two English sentences are rarely combined into one Hindi sentence (when translating from English to Hindi), whereas the reverse is quite possible.

8 Evaluation Results

The results for various corpus types are given in table-1, for corpus sizes in table-2, and the global measures in table-3. Among the four algorithms tested, Moore's (**Mre**) gives the best results (except for the EMILLE corpus). This is as expected, since **Mre** combines sentence length based method with word correspondence. The results for **Mmd** are the worst, but it should be noted that the results for **Mmd** reported in this paper may not be the best that can be obtained with it, because its performance depends on some parameters. Perhaps with better tuning for English-Hindi, it might perform better. Another expected outcome is that the results for **GC** (character count) are better than **Brn** (word count). One reason for this is that there are more of characters than words (Gale and Church, 1991).

Leaving aside the tuning aspect, the low performance of **Mmd** may be due to the fact that it relies on cognate matching, and there are fewer cognates between Hindi and English. It might also be due to the syntactic differences (word order) between Hindi and English. This could, perhaps be taken care of by increasing the maximum point dispersal threshold (relaxing the linearity constraint), as suggested by Melamed (Melamed, 1996).

The results of experiment on English-French (table-1) show that **Mmd** performs better for this language pair than for English-Hindi, but it still seems to be more sensitive to noise than the other three algorithms. **Mre** performed the best for English-French too.

With respect to speed, **Brn** and **GC** are the fastest, **Mre** is marginally slower, and **Mmd** is much slower.

The effects of the previously mentioned factors on performance have been summarized below.

8.1 Corpus Type

Brn, **GC**, and **Mmd** performed almost equally well for EMILLE and ERDC corpora, but not that well for India Today. However, surprisingly, **Mre** performed much worse for EMILLE than it did for the other two corpora. It could be because of the fact that the EMILLE has a lot of very short (1-3 words) sentences, and word correspondence (in the second pass) may not be that effective for such sentences. The results don't support our assumption that EMILLE is easier than ERDC, but India Today does turn out to be more difficult than the other two for all the test cases. This is understandable since the translations in this corpus are much less literal.

8.2 Corpus Size

Only in the case of **Mre**, the performance almost consistently increased with size. This is as expected since the second pass in **Mre** needs training from the results of the first pass. The corpus size has to be large for this training to be effective. There doesn't seem to be a clear relationship between size and performance for the other three algorithms.

8.3 Noise and Difference in Sizes of SL and TL Corpora

As expected, introducing noise led to a decrease in performance for all the algorithms (table-1 and table-2). However (barring EMILLE) **Mre** seems to become less sensitive to noise as the corpus size increases. This again could be due to the unsupervised learning aspect of **Mre**.

Making the SL and TL corpora differ in size tended to reduce the performance in most cases, but sometimes the performance marginally improved.

Table 3: Global Evaluation Measures

		Brn	**GC**	**Mmd**	**Mre**
Clean, Same Size	L	92.6	93.4	81.4	80.8
	H	100.0	100.0	96.3	100.0
	P	98.4	98.7	90.3	95.1
	R	96.1	96.1	87.6	90.0
	F	97.2	97.3	88.9	92.4
Noisy, Same Size	L	73.1	75.8	44.1	72.6
	H	87.5	86.4	62.4	92.3
	P	82.7	84.1	53.8	92.2
	R	77.4	78.4	52.8	74.9
	F	79.8	81.1	53.3	82.5
Noisy, Different Size	L	74.7	76.4	46.2	71.3
	H	85.6	86.4	55.0	92.0
	P	83.4	84.9	51.2	91.5
	R	77.2	78.3	50.0	74.0
	F	80.1	81.4	50.6	81.6
Overall	L	81.1	82.4	55.4	80.0
	H	90.4	90.8	73.1	91.0
	P	88.2	89.2	65.1	92.9
	R	83.6	84.3	63.5	79.6
	F	85.7	86.6	64.6	85.5

L and *H:* Lower and higher limits of 95% confidence interval for F-measure
P, *R*, and *F:* Average precision, recall, and F-measure

9 Some Notes on Actual Corpus Alignment

Based on the evaluation results and our experience while manually checking alignments, we make some observations below which could be useful to those who are planning to create aligned parallel corpora.

Contrary to what we believed, sentence length based algorithms turn out to be quite robust, but also contrary to the commonly held view, there is scope for improvement in the performance of these algorithms by combining them with other techniques as Moore has done. However, as the performance of **Mre** on EMILLE shows, these additional techniques might sometimes *decrease* the performance.

There is a tradeoff between precision and recall, just as between robustness and accuracy (Simard and Plamondon, 1998). If the corpus aligned automatically is to be used without manual checking, then we should opt for maximum precision. But if it's going to be manually checked before being used, then we

should opt for maximum recall. It depends on the application too (Langlais et al., 1996), but if manual checking is to be done, we can as well try to get the maximum number of alignments, since some decrease in precision is not going to make manual checking much more difficult.

If the automatically aligned corpus is not to be checked manually, it becomes even more important to perform a systematic evaluation before aligning a corpus, otherwise the parallel corpus will not be reliable either for machine learning or for linguistic analysis.

10 Conclusion

We used a systematic evaluation method for selecting a sentence alignment algorithm with English and Hindi as the language pair. We tested four algorithms for different corpus types and sizes, for the same and different sizes of SL and TL corpora, as well as presence and absence of noise. The evaluation scheme we have described can be used for a more meaningful comparison of sentence alignment algorithms. The results of the evaluation show that the performance depends on various factors. The direction of this variation (increase or decrease) was as predicted in most of the cases, but some results were unexpected. We also presented some suggestions on using an algorithm for actual alignment.

References

Brown Peter F., Cocke John, Della Pietra Stephen A., Della Pietra Vincent J., Jelinek Frederick, Lafferty John D., Mercer Robert L., and Roossin Paul S. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics.*

Brown Peter F., Della Pietra Stephen A., Della Pietra Vincent J., and Mercer Robert L. 1993. Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Brown Peter F., Lai J. C. and Mercer Robert L. 1991. Aligning Sentences in Parallel Corpora. *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, 169–176. Berkeley, CA.

Chen Stanley F. 1993. Aligning Sentences in Bilingual Corpora Using Lexical Information. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 9–16. Columbus, OH.

Church Kenneth W. 1993. Char_align: A Program for Aligning Parallel Texts at the Character Level. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1–8. Columbus, OH.

Church Kenneth W. and Hanks Patrick. 1993b. Aligning Parallel Texts: Do Methods Developed for English-French Generalize to Asian Languages?. *Proceedings of Rocling.*

Gale William A. and Church Kenneth W. 1991. A Program for Aligning Sentences in Bilingual Corpora. *Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, 177–184. Berkeley, CA.

Kay Martin. 1991. Text-Translation Alignment. *ACH/ALLC '91: "Making Connections" Conference Handbook.* Tempe, Arizona.

Kay Martin and Roscheisen Martin. 1993. Text-Translation Alignment. *Computational Linguistics*, 19(1):121–142.

Langlais Phillippe, Simard Michel, and Vronis Jean. 1996. Methods and Practical Issues in Evaluating Alignment Techniques. *Proceedings of 16th International Conference on Computational Linguistics (COLING-96).*

Melamed I. Dan. 1996. A Geometric Approach to Mapping Bitext Correspondence. *IRCS Technical Report, University of Pennsylvania*, 96–22.

Moore Robert C. 2002. Fast and Accurate Sentence Alignment of Bilingual Corpora. *Proceedings of AMTA*, 135–144.

Och Franz Joseph and Ney Hermann 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19-51.

Simard Michel, Foster George F., and Isabelle Pierre. 1992 Using Cognates to Align Sentences in Bilingual Corpora. *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation.* Montreal, Canada.

Simard Michel and Plamondon Pierre. 1998 Bilingual Sentence Alignment: Balancing Robustness and Accuracy. *Machine Translation*, 13(1):59–80.

Wu Dekai. 1994. Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria. *Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics*, 80–87. Las Cruces, NM.

Combined word alignments

Dan Tufiş, Radu Ion, Alexandru Ceauşu, Dan Ştefănescu
Romanian Academy Institute for Artificial Intelligence
13, "13 Septembrie", 74311, Bucharest 5, Romania
{tufis, radu, alceusu, danstef}@racai.ro

Abstract

We briefly describe a word alignment system that combines two different methods in bitext correspondences identification. The first one is a hypotheses testing approach (Gale and Church, 1991; Melamed, 2001; Tufiş 2002) while the second one is closer to a model estimating approach (Brown et al., 1993; Och and Ney, 2000). We show that combining the two aligners the results are significantly improved as compared to each individual aligner.

Introduction

In (Tufiş, 2002) we described a translation equivalence extraction program called TREQ the development of which was twofold motivated: to help enriching the synsets of the Romanian wordnet (Tufiş et al. 2004a) with new literals based on bilingual corpora evidence and to check the interlingual alignment of our wordnet against the Princeton Wordnet. The translation equivalence extractor has been also incorporated into a WSD system (Tufiş et al., 2004b) part of a semantic web annotation platform. It also constituted the backbone of our **TREQ-AL** word aligner which successfully participated in the previous HLT-NAACL 2003 Shared Task[1] on word alignment for Romanian-English parallel texts. A detailed description of TREQ&TREQ-AL is given in (Tufiş et al. 2003b) and it will be very shortly overviewed.

A quite different approach from our hypotheses testing implemented in the TREQ-AL aligner is taken by the model-estimating aligners, most of them relying on the IBM models (1 to 5) described in the (Brown et al. 1993) seminal paper. The first wide-spread and publicly available implementation of the IBM models was the GIZA program, which itself was part of the SMT toolkit EGYPT (Al-Onaizan et al., 1999). GIZA has been superseded by its recent extension GIZA++ (Och and Ney, 2000, 2003) publicly available[2]. We used the translation probabilities generated by GIZA++ for implementing a second aligner, **MEBA**, described in a little more details in a subsequent section. The alignments produced by MEBA were compared to the ones produced by TREQ-AL. We used for comparison the Gold Standard[3] annotation from the HLT-NAACL 2003 Shared Task. In order to combine the two aligners we had to check whether their accuracy was comparable and that when they are wrong the set of mistakes made by one aligner is not a proper set of the errors made by the second one. The first check was performed by using McNamer's test (Dieterich, 1998) and for the second we used Brill &Wu test (Brill, Wu, 1998). Both tests confirmed that the conditions for combining were ensured so, we built the combiner.

The Combined Word Aligner, **COWAL**, is a wrapper of the two aligners (TREQ-AL and MEBA) ensuring the pre- and post-processing. It is complemented by a graphical user interface that allows for the visualisation of the alignments (intermediary and the final ones) as well as for their editing. We should note that the corrections made by the user are stored by COWAL as positive and negative examples for word dependencies (in the monolingual context) and translation equivalencies (in the bilingual context). In the current version the editorial logs are used by the human developers but we plan to further extend COWAL for automatic learning from this extremely valuable kind of data.

The bitext processing

The two base aligners and their combination use the same format for the input data and provide the alignments in the same format. The input format is obtained from two raw texts which represent reciprocal translations. If not already sentence aligned, the two texts are aligned. In the shared task this step was not necessary since both the training data and evaluation data were provided in the sentence aligned format.

The texts in each language are then tokenized with the MULTEXT multilingual tokenizer[4]. The tokenizer is a finite state automaton using language specific

[1] http://www.cs.unt.edu/~rada/wpt/index.html#shared
[2] http://www.fjoch.com/GIZA++.2003-09-30.tar.gz

[3] We noticed in the Gold Standard two sentences where alignments were wrongly shifted by one position (due to an unprintable character) and we corrected them.
[4] http://aune.lpl.univ-aix.fr:16080/projects/multext/MtSeg/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 107–110,
Ann Arbor, June 2005.

resources. It recognizes several compounds (phrasal verbs, idioms, dates) and split contrasted or cliticized constructions. This tokenization considerably differs from the one prescribed by the Shared Task where a token is any character string delimited by a blank or a punctuation sign (which itself is considered a token).

Since our processing tools (especially the tokeniser) were built with a different segmentation strategy in mind, we generated the alignments based on our own tokenization and, at the end, we "re-tokenised" the text according to original evaluation data (and consequently re-index) all the linking pairs. After tokenization, both texts are tagged and lemmatized. We used in-house language models and lemmatizers and the Brants's TnT tagger[5]. For both English and Romanian we used MULTEXT-EAST[6] compliant tagsets. With different tags, a tagset mapping table becomes an obligatory external resource. Although, more often than not, the translation equivalents have the same part-of speech, relying on such a restriction would seriously affect the alignment recall. However, when the translation equivalents have different parts of speech, this difference is not arbitrary. During the training phase we estimated bilingual *POS affinities:* $\{p(POS_m^{RO}| POS_n^{EN})\}$ and $\{p(POS_n^{EN}|POS_m^{RO})\}$. POS affinities were used as one of the information sources in dealing with competitive alignments.

The next preprocessing step is represented by a rather primitive form of sentence chunking in both languages. They roughly correspond to (non-recursive) noun phrases, adjectival phrases, prepositional phrases and verb complexes (analytical realization of tense, aspect mood and diathesis and phrasal verbs). The "chunks" are recognized by a set of regular expressions defined over the tagsets. Finally, the bitext is assembled as an XML document (XCES-Align-ana format), as used in the MULTEXT-EAST corpus, which is the standard input for most of our tools, including COWAL alignment platform.

The three aligners

TREQ-AL generates translation equivalence hypotheses for the pairs of words (one for each language in the parallel corpus) which have been observed occurring in aligned sentences more than expected by chance. The hypotheses are filtered by a loglikelihood score threshold. Several heuristics (string similarity-cognates, POS affinities and alignments locality[7]) are used in a competitive linking manner (Melamed, 2001) to make the final decision on the most likely translation equivalents. Given that, initially, this program was designed for extracting translation equivalents for the alignment of the Romanian wordnet to the Princeton wordnet, it deals only with one to one mappings. To cope with the many to many mappings (especially for functional words alignment), the earlier version of the translation equivalence extractor encoded some general rules assumed to be valid over a large set of natural languages such as: auxiliaries and verbal particles (infinitive, subjunctive, aspectual and temporal) are related to the closest main verb, determiners (articles, pronominal adjectives, quantifiers) are related to the closest nominal category (noun or pronoun). Currently this part of the TREQ-AL code became redundant because the chunking module mentioned before does the same job in a more general and flexible way.

MEBA is an iterative algorithm which uses the translation probabilities, distorsions and POS-affinities generated by GIZA++ and takes advantage of all preprocessing phases mentioned in the previous section. In each step are aligned different categories of tokens (content words, named entities, functional words) in decreasing order of statistical evidence. The score of a link is computed by a linear function of 7 parameters' scores: translation probability, POS affinity, string similarity, *alignments locality* (both strict and weaker versions) distortions and the entropy of the translation equivalents. For all these parameters, in each processing step, we empirically set minimal thresholds and various weights. The tokens considered for the computing translation probabilities are the lemmas trailed by the grammatical categories (eg. plane_N, plane_V plane_A). This way we aimed at avoiding data sparseness and filtering noisy data. For highly inflectional languages (as Romanian is) the use of lemmas instead of word occurrences contributes significantly to the data sparseness reduction. For languages with weak inflectional character (as English is) the POS trailing contributes especially to the filtering the search space. Each processing step is controlled by above mentioned parameters, the weights and thresholds of which vary from step to step (even the order of the processing steps is one of the possible parameters).

The first alignment step builds only links with a high level of certainty (that is cognates, pairs of high translation probability and high POS affinity). The grammatical categories which are considered in this step are user controlled (usually nouns, adjectives or non-auxiliary verbs and which have the fewest competitive translations). The next processing steps try to align

[5] http://acl.ldc.upenn.edu/A/A00/A00-1031.pdf

[6] http://nl.ijs.si/ME/V2/

[7] The *alignments locality* heuristics exploits the observation made by several researchers that adjacent words of a text in the source language tend to align to adjacent words in the target language. A more strict alignment locality constraint requires that all alignment links starting from a chunk, in the one language end in a chunk in the other language. This restricted form of locality is relevant for related languages.

content words (open class categories) as confidently as possible, following the alignments in previous steps as anchor points. In all steps the candidates are considered if and only if they meet the minimal threshold restrictions. If the input bitext is chunked, the strict alignment locality heuristics is very effective to determine the correct alignment even for unseen pairs of words (or for which the translation equivalence probability is below the considered threshold). When the pre-chunking of the parallel texts is not available, MEBA uses the weaker form of the locality heuristics by analyzing the alignments already existing in a window of N tokens centered on the focused token. The window size is variable, proportional to the sentence length. For all alignments in the window, an average displacement is computed and, among the competing alignments, preference will be given to the links with displacement values closer to the average one.

The functional words and punctuation are processed in the last step and their alignments are guided by the POS-affinities and alignment locality heuristics. If none of the alignment clues or their combination (Tiedemann, 2003) is strong enough, the functional words are automatically aligned with the word(s) their governor is aligned to. The governor is chunk-based defined: it is the content word of a chunk (if there are more content words in a chunk, then the governor is the grammatical head). If the chunking is not available, the closest content word is selected as the governor. Proximity is checked to the left or to the right according to the frequencies of the POS-ngram containing the current functional word.

We should mention that the probabilities computed during the training phase are not re-estimated for each run-time processing step. At run-time only the weights and thresholds change from step to step.

***COWAL*,** the combined aligner takes advantage of the alignments independently provided by TREQ-AL and MEBA. The simplest combination method consists in computing either the union (high recall, low precision), or the intersection (lower recall, higher precision) of the independent alignments. We evaluated both these simple methods of combination and found that the best F-measure was provided by the union-based combination. Although for the shared task we submitted the union-based combined alignment (*Baseline COWAL*, see Table 1), there are various ways to improve it. We discuss three cases where improvement is possible (C1, C2 and C3, see below) and which were evaluated after the submission deadline. The results of this (unofficial) evaluation are summarized in Table 1 by the *f-COWAL* line. These cases refer to competing links that appeared after the union of the independent alignments. The conflicts resolution is based on the (weak) locality and distortion heuristics discussed before. The currently identified competing links are only those for which the following conditions apply:

C1) if one aligner found for a word W a non-null alignment and the other aligner generated for the same word W a null link, then the baseline alignment contains an impossible situation: the token W is recorded both as translated and not-translated in the other language. The translation probabilities, POS affinity and the relative displacement of the tokens in the non-null candidates were the strongest decision criteria. We found that in about 60% of the cases the null alignments were mistaken. So, for the time being, we simply eliminated the null competing alignments (this should be addressed in a more principled way by the future version of the combiner).

C2) long distant competing links; this case appears when one aligner found for the word Ws the link to the target word Wt_m, the other aligner found for Ws the target Wt_n, and the distance between Wt_m and Wt_n, is more than 3 words (in a future version this maximum distance will be a dynamic parameter, depending on the sentence length and the POS of Ws).

C3) competing links to the same target(s) of a word occurring several times in the same sentence; consider, for example, the Romanian fragment:
"…la_1 Neptun, la_2 Orastie si la_3 Afumati, …
which in English is translated by the next segment:
"…in Neptun, Orastie and Afumati…
In spite of the gold standard considering that all three occurrences of the preposition "la" in Romanian (la_1, la_2 ,la_3) are aligned to the same word in English ("in"), the filtering, in this case, licensed only the alignment "la_1 <-> in". We consider that this filtered alignment is correct, since omitting "la_2" and "la_3" does not alter the syntactic correctness of the Romanian text, and also because the insertion in the English fragment of the preposition "in" before "Orastie" and before "Afumati" wouldn't alter the grammaticality of the English fragment. Since both repetitions and omissions are optional, we consider that only the first occurrence of the preposition ("la_1") is translated in English, while the others are omitted.

Another possible improvement (not implemented yet) was revealed by observing that the final result contained several incomplete n-m (phrasal) alignments. It is likely that even an elementary n-gram analysis (both sides of the bitext) would bring valuable evidence for improving the phrasal alignments.

Post-processing

As said in the second section, our tokenization was different from the tokenization in the training and test data. To comply with the evaluation protocol, we had to re-tokenize the aligned text and re-compute the indexes

of the links. Some multi-word expressions recognized by the tokenizer as one token, such as dates (*25 ianuarie, 2001*), compound prepositions (*de la*, *până la*), conjunctions (*pentru ca*, *de când*, *până când*) or adverbs (*de jur împrejur*, *în față*) as well as the hyphen separated nominal compounds (*mass-media*, *prim-ministru*) were split, their positions were re-indexed and the initial one link of a split compound was replaced with the set obtained by adding one link for each constituent of the compound to the target English word. The same hold for the other way around. Therefore if two multiword expressions were initially found to be translation equivalents (one alignment link) after the post-processing number of generated links became N*M, where N represented the number of words in the first language compound and M the number of words in the second language compound.

Evaluation and conclusions

Neither TREQ-AL nor MEBA needs an a priori bilingual dictionary, as this will be automatically extracted by the TREQ or GIZA++. We made evaluation of the individual alignments in both experimental settings: without a startup bilingual lexicon and with an initial mid-sized bilingual lexicon. Surprisingly enough, we found that while the performance of TREQ-AL increases a little bit (approx. 1% increase of the F-measure) MEBA is doing better without an additional lexicon. So, in the evaluation below MEBA uses only the training data vocabulary.

Aligner	Precision	Recall	F-meas.	AER
TREQ-AL	81.71	60.57	69.57	30.43
MEBA	**82.85**	60.41	69.87	30.13
Baseline (union)COWAL	70.84	**76.67**	**73.64**	26.36
f-COWAL (H1+H2+H3)	**87.17**	70.25	**77.80**	**22.20**

Table 1. Evaluation results against the official GS

After the release of the official Gold Standard we noticed and corrected some obvious errors and also removed the controversial links of the type c) discussed in the previous section. The evaluations against this new "Gold Standard" showed, on average, 3.5% better figures (precision, recall, F-measure and AER) for the individual aligners, while for the combined classifiers, the performance scores were about 4% better.

MEBA is very sensitive to the values of the parameters which control its behavior. Currently they are set according to the developers' intuition and after the analysis of the results from several trials. Since this activity is pretty time consuming (human analysis plus re-training might take a couple of hours) we plan to extend MEBA with a supervised learning module, which would automatically determine the "optimal" parameters (thresholds and weights) values.

References

Al-Onaizan, Y., Curin, J., Jahr, M., Knight K., Lafferty, J., Melamed, D., Och, F. J., Purdy, D., Smith, N.A., Yarowsky, D. (1999) : Statistical Machine Translation, Final Report, JHU Workshop, 42 pages

Brill, E., and Wu, J. (1998). "Classifier Combination for Improved Lexical Disambiguation" *In Proceedings of COLING-ACL'98* Montreal, Canada, 191-195

Brown, P. F., Della Pietra, S.A., Della Pietra, V. J., Mercer, R. L.(1993) "The mathematics of statistical machine translation: Parameter estimation". *Computational Linguistics*, 19(2) pp. 263–311.

Dietterich, T. G., (1998). "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". *Neural Computation, 10* (7) 1895-1924.

Gale, W.A. and Church, K.W. (1991). „Identifying word correspondences in parallel texts". Proceedings of the Fourth DARPA Workshop on Speech and Natural Language. Asilomar, CA, pp. 152–157.

Melamed, D. (2001). *Empirical Methods for Exploiting Parallel Texts*. Cambridge, MA: MIT Press.

Och, F.J., Ney, H. (2003) "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, 29(1), pp. 19-51

Och, F.J., Ney, H.(2000) "Improved Statistical Alignment Models". Proceedings of the 38th ACL, Hongkong, pp. 440-447

Tiedemann, J. (2003) "Combining clues for word alignment". In Proceedings of the 10th EACL, Budapest, pp. 339–346

Tufiş, D.(2002) "A cheap and fast way to build useful translation lexicons". Proceedings of COLING2002, Taipei, pp. 1030-1036.

Tufiş, D., Barbu, A.M., Ion R (2003).: „TREQ-AL: A word-alignment system with limited language resources", Proceedings of the NAACL 2003 Workshop on Building and Using Parallel Texts; Romanian-English Shared Task, Edmonton, pp. 36-39

Tufiş, D., Ion, R., Ide, N.(2004a): Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets. Proceedings of COLING2004, Geneva, pp. 1312-1318

Tufis, D., Barbu, E., Mititelu, V., Ion, R., Bozianu, L.(2004b): „The Romanian Wordnet". In Romanian Journal on Information Science and Technology, Dan Tufiş (ed.) Special Issue on BalkaNet, Romanian Academy, 7(2-3), pp. 105-122.

LIHLA: Shared task system description

Helena M. Caseli, Maria G. V. Nunes
NILC – ICMC – Univ. São Paulo
CP 668P, 13560-970 São Carlos–SP, Brazil
{helename,gracan}@icmc.usp.br

Mikel L. Forcada
Transducens – DLSI – Univ. d'Alacant
E-03071 Alacant, Spain
mlf@dlsi.ua.es

Abstract

In this paper we describe LIHLA, a lexical aligner which uses bilingual probabilistic lexicons generated by a freely available set of tools (NATools) and language-independent heuristics to find links between single words and multiword units in sentence-aligned parallel texts. The method has achieved an alignment error rate of 22.72% and 44.49% on English–Inuktitut and Romanian–English parallel sentences, respectively.

1 Introduction

Alignment of words and multiword units plays an important role in many natural language processing (NLP) applications, such as example-based machine translation (EBMT) (Somers, 1999) and statistical machine translation (SMT) (Ayan et al., 2004; Och and Ney, 2000), transfer rule learning (Carl, 2001; Menezes and Richardson, 2001), bilingual lexicography (Gómez Guinovart and Sacau Fontenla, 2004), and word sense disambiguation (Gale et al., 1992), among others.

Aligning two (or more) texts means finding correspondences (translation equivalences) between segments (paragraphs, sentences, words, etc.) of the source text and segments of its translation (the target text). Following the same idea of many recently proposed approaches on lexical alignment (e.g., Wu and Wang (2004) and Ayan et al. (2004)), the method described in this paper, LIHLA (Language-Independent Heuristics Lexical Aligner) starts from statistical alignments between single words (defined in bilingual lexicons) and applies language-independent heuristics to them, aiming at finding the best alignments between words or multiword units.

Although the most frequent alignment category is $1:1$ (in which one source word is translated exactly as one target word), other categories such as omissions ($1:0$ or $0:1$) or those involving multiword units ($n:m$, with n and/or $m \geq 1$) are also possible.

This paper is organized as follows: section 2 explains how LIHLA works; section 3 describes some experiments carried out with LIHLA together with their results and, in section 4, some concluding remarks are presented.

2 How LIHLA works

As the first step, LIHLA uses alignments between single words defined in two bilingual lexicons (source–target and target–source) generated from sentence-aligned parallel texts using NATools.[1]

Given two sentence-aligned corpus files, the NATools word aligner —based on the Twenty-One system (Hiemstra, 1998)— counts the co-occurrences of words in all aligned sentence pairs and builds a sparse matrix of word-to-word probabilities (Model A) using an iterative expectation-maximization algorithm (5 iterations by default). Finally, the elements with higher values in the matrix are chosen to compose two probabilistic bilingual lexicons (source–target and target–source) (Simões and Almeida, 2003). For each word in the corpus, each

[1] NATools is a set of tools developed to work with parallel corpora, which is freely available in http://natura.di.uminho.pt/natura/natura/.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 111–114,
Ann Arbor, June 2005.

bilingual lexicon gives: the number of occurrences of that word in the corpus (its absolute frequency) and its most likely translations together with their probabilities.

The construction of the bilingual lexicons is an independent prior step for the alignment performed by LIHLA and the same bilingual lexicons can be used several times to align parallel sentences.

So, using the two bilingual lexicons generated by NATools and some language-independent heuristics, LIHLA tries to find the best alignment between source and target tokens (words, numbers, special characters, etc.) in a pair of parallel sentences. For each source token s_j in source sentence S, LIHLA will look for the best token t_i in the target parallel sentence T applying these heuristics in sequence:

1. **Exact match**
   LIHLA creates a $1:1$ alignment between s_j and t_i if they are identical. This heuristic stays for exact matches, for instance, between proper names and numbers.

2. **Best candidate according to the bilingual lexicon**
   LIHLA looks for possible translations of s_j in the source–target bilingual lexicon (B_S) and makes an intersection between them and the words in T. In this intersection, if no candidate word identical to those in B_S is found, then LIHLA tries to look for cognates for those words using the longest common subsequence ratio (LCSR).[2] By doing this, LIHLA can deal with small changes in possible translations such as different forms of the same verb, changes in gender and/or number of nouns, adjectives, and so on.
   Then, LIHLA selects the best target candidate word t_i for s_j —the best candidate word according to B_S among those in a position which is favorably situated in relation to s_j— and looks for multiword units involving s_j and t_i —those words that occur immediately before and/or after s_j (for source multiword units) or t_i (for target multiword units) and are not possible translations for other words in T and S, respectively. According to the multiword units that have (or not) been found, a $1:1$, $1:n$, $m:1$ or $m:n$ alignment is established. An omission alignment for s_j ($1:0$) can also be established if no target candidate word t_i that satisfies this heuristic is available.

3. **Cognates**
   If no possible translation for s_j is found in the bilingual lexicon and the target sentence (T) at the same time, LIHLA uses the LCSR to look for cognates for s_j in T and sets a $1:1$ alignment between s_j and its best cognate or a $1:0$ alignment if there is no cognate available.

These heuristics are applied while alignments can still be produced and a maximum number of iterations is not reached (see section 3 for the number of iterations performed in the experiments described in this paper). Furthermore, at the first iteration, all words with a frequency higher than a set threshold are ignored to avoid erroneous alignments since all subsequent alignments are based on the previous ones.

In its last step (which is optional and has not been performed in the experiments described in this paper), LIHLA aligns the remaining unaligned source and target tokens between two pairs of already aligned tokens establishing several $1:1$ alignments when there are the same number of source and target tokens, or just one alignment involving all source and target tokens if they exist in different quantities. The decision of creating n $1:1$ alignments in spite of just one $n:n$ alignment when there is the same number of source and target tokens is due to the fact that a $1:1$ alignment is more likely to be found than a $n:n$ one.

3 Experiments

In this section we present the experiments carried out with LIHLA for the "Shared task on word alignment" in the Workshop on Building and Using Parallel Texts during ACL2005. Systems participating in this shared task were provided with training data (consisting of sentence-aligned parallel texts) for three pairs of languages: English–Inuktitut,

[2] The LCSR of two words is computed by dividing the length of their longest common subsequence by the length of the longer word. For example, the LCSR of Portuguese word *alinhamento* and Spanish word *alineamiento* is $\frac{10}{12} \simeq 0.83$ as their longest common subsequence is *a-l-i-n-a-m-e-n-t-o*.

Romanian–English and English–Hindi. Furthermore, the systems would choose to participate in one or both subtasks of "limited resources" (where systems were allowed to use only the resources provided) and "unlimited resources" (where systems were allowed to use any resources in addition to those provided). The system described in this paper, LIHLA, participated in the subtask of limited resources aligning English–Inuktitut and Romanian–English test sets.

The training sets —composed of 338,343 English–Inuktitut aligned sentences (omission cases were excluded from the whole set of 340,526 pairs) and 48,478 Romanian–English aligned ones— were used to build the bilingual lexicons. Then, without changing any default parameter (threshold for LCSR, maximum number of iterations, etc.), LIHLA aligned the 75 English–Inuktitut and the 203 Romanian–English parallel sentences on test sets. The whole alignment process (bilingual lexicon generation and alignment itself) did not take more than 17 minutes for English–Inuktitut (3 iterations per sentence, on average) and 7 minutes for Romanian–English (4 iterations per sentence, on average).

The evaluation was run with respect to precision, recall, F-measure, and alignment error rate (AER) considering sure and probable alignments but not NULL ones (Mihalcea and Pedersen, 2003). Tables 1 and 2 present metric values for English–Inuktitut and Romanian–English alignments, respectively, as provided by the organization of the shared task.

Metric	Sure	Probable
Precision	46.55%	79.53%
Recall	73.72%	18.71%
F-measure	57.07%	30.30%
AER	22.72%	

Table 1: LIHLA results for English–Inuktitut

Metric	Sure	Probable
Precision	57.68%	57.68%
Recall	53.51%	53.51%
F-measure	55.51%	55.51%
AER	44.49%	

Table 2: LIHLA results for Romanian–English

The results obtained in these experiments were not so good as those achieved by LIHLA on the language pairs for which it was developed, that is, 92.48% of precision and 88.32% of recall on Portuguese–Spanish parallel texts and 84.35% of precision and 76.39% of recall on Portuguese–English ones.[3]

The poor performance in the English–Inuktikut task may be partly due to the fact that Inuktikut is a polysynthetic language, that is, one in which, unlike in English, words are formed by long strings of concatenated morphemes. This makes it difficult for NATools to build reasonable dictionaries and lead to a predominance of $n : 1$ alignments, which are harder to determine —this fact can be confirmed by the better precision of LIHLA when probable alignments were considered (see table 1). The performance in the English–Romanian task, not very far from the English–Portuguese task used to tune up the parameters of the algorithm, is harder to explain without further analysis.

The difference in precision and recall between the two language pairs is due to the fact that on the English–Inuktitut reference corpus in addition to sure alignments the probable ones were also annotated while in Romanian–English only sure alignments are found. This indicates that evaluating alignment systems is not a simple task since their performance depends not only on the language pairs and the quality of parallel corpora (constant criteria in this shared task) but also the way the reference corpus is built.

So, at this moment, it would be unfair to blame the worse performance of LIHLA on its alignment methodology since it has been applied to the new language pairs without changing any of its default parameters. Maybe a simple optimization of parameters for each pair of languages could bring better results and also the impact of size and quality of training and reference corpora used in these experiments should be investigated. Then, the only conclusion that can be taken at this moment is that LIHLA, with its heuristics and/or default parameters, can not be indistinctly applied to any pair of languages.

Despite of its performance, LIHLA has some

[3]For more details of these experiments see (Caseli et al., accepted paper).

advantages when compared to other lexical alignment methods found in the literature, such as: it does not need to be trained for a new pair of languages (as in Och and Ney (2000)) and neither does it require pre-processing steps to handle texts (as in Gómez Guinovart and Sacau Fontenla (2004)). Furthermore, the whole alignment process (bilingual lexical generation and alignment itself) has proved to be very fast as mentioned previously.

4 Concluding remarks

This paper has presented a lexical alignment method, LIHLA, which aligns words and multiword units based on initial statistical word-to-word correspondences and language-independent heuristics.

In the experiments carried out at the "Shared task on word alignment" which took place at the Workshop on Building and Using Parallel Texts during ACL2005, LIHLA has been evaluated on English–Inuktitut and Romanian–English parallel texts achieving an AER of 22.72% and 44.49%, respectively.

As future work, we aim at investigating the impact of using additional linguistic information (such as part-of-speech tags) on LIHLA's performance. Also, as a long-term goal, LIHLA will be part of a system implemented to learn transfer rules from sequences of aligned words.

Acknowledgments

We thank FAPESP, CAPES, CNPq and the Spanish Ministry of Science & Technology (Project TIC2003-08681-C02-01) for financial support.

References

Necip F. Ayan, Bonnie J. Dorr, and Nizar Habash. 2004. Multi-Align: Combining linguistic and statistical techniques to improve alignments for adaptable MT. In R. E. Frederking and K. B. Taylor, editors, *Proceedings of the 6th Conference of the AMTA (AMTA-2004)*, number 3265 in Lecture Notes in Artificial Inteligence (LNAI), pages 17–26. Springer-Verlag Berlin Heidelberg.

Michael Carl. 2001. Inducing probabilistic invertible translation grammars from aligned texts. In *Proceedings of CoNLL-2001*, pages 145–151, Toulouse, France.

Helena M. Caseli, Maria das Graças V. Nunes, and Mikel L. Forcada. (accepted paper). LIHLA: A lexical aligner based on language-independent heuristics. In *Proceedings of the V Encontro Nacional de Inteligência Artificial (ENIA05)*, São Leopoldo, RS, Brazil.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1992)*, pages 101–112, Montreal, Canada, June.

Xavier Gómez Guinovart and Elena Sacau Fontenla. 2004. Métodos de optimización de la extracción de léxico bilingüe a partir de corpus paralelos. *Procesamiento del Lenguaje Natural*, 33:133–140.

Djoerd Hiemstra. 1998. Multilingual domain modeling in Twenty-One: automatic creation of a bi-directional translation lexicon from a parallel corpus. In Peter Arno Coppen, Hans van Halteren, and Lisanne Teunissen, editors, *Proceedings of the 8th CLIN meeting*, pages 41–58.

Arul Menezes and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the Workshop on Data-driven Machine Translation at 39th Annual Meeting of the ACL (ACL-2001)*, pages 39–46, Toulouse, France.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts Data Driven Machine Translation and Beyond*, pages 1–10, Edmonton, May–June.

Franz J. Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL (ACL-2000)*, pages 440–447, Hong Kong, China, October.

Alberto M. Simões and José J. Almeida. 2003. NATools – A statistical word aligner workbench. *Procesamiento del Lenguaje Natural*, 31:217–224.

Harold Somers. 1999. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157.

Hua Wu and Haifeng Wang. 2004. Improving domain-specific word alignment with a general bilingual corpus. In R. E. Frederking and K. B. Taylor, editors, *Proceedings of the 6th Conference of the AMTA (AMTA-2004)*, number 3265 in Lecture Notes in Artificial Inteligence (LNAI), pages 262–271. Springer-Verlag Berlin Heidelberg.

Aligning words in English-Hindi parallel corpora

Niraj Aswani
Department of Computer Science
University of Sheffield
Regent Court 211, Portobello Street
Sheffield S1 4DP, UK
N.Aswani@dcs.shef.ac.uk

Robert Gaizauskas
Department of Computer Science
University of Sheffield
Regent Court 211, Portobello Street
Sheffield S1 4DP, UK
R.Gaizauskas@dcs.shef.ac.uk

Abstract

In this paper, we describe a word alignment algorithm for English-Hindi parallel data. The system was developed to participate in the shared task on word alignment for languages with scarce resources at the ACL 2005 workshop, on "Building and using parallel texts: data driven machine translation and beyond". Our word alignment algorithm is based on a hybrid method which performs local word grouping on Hindi sentences and uses other methods such as dictionary lookup, transliteration similarity, expected English words and nearest aligned neighbours. We trained our system on the training data provided to obtain a list of named entities and cognates and to collect rules for local word grouping in Hindi sentences. The system scored 77.03% precision and 60.68% recall on the shared task unseen test data.

1 Introduction

This paper describes a word alignment system developed as a part of shared task on word alignment for languages with scarce resources at the ACL 2005 workshop on "building and using parallel texts: data driven machine translation and beyond". Participants in the shared task were provided with common sets of training data, consisting of English-Inuktitut, Romanian-English, and English-Hindi parallel texts and the participating teams could choose to evaluate their system on one, two, or all three language pairs. Our system is for aligning English-Hindi parallel data at the word level. The word-alignment algorithm described here is based on a hybrid – multi-feature approach, which groups Hindi words locally within a Hindi sentence and uses dictionary lookup (DL) as the main method of aligning words along with other methods such as Transliteration Similarity (TS), Expected English Words (EEW) and Nearest Aligned Neighbors (NAN). We used the training data supplied to derive rules for local word grouping in Hindi sentences and to find Named Entities (NE) and cognates using our TS approach. In the following sections we briefly describe our approach.

2 Training Data

The training data set was composed of approximately 3441 English-Hindi parallel sentence pairs drawn from the EMILLE (Enabling Minority Language Engineering) corpus (Baker et al., 2004). The data was pre-tokenized. For the English data, a token was a sequence of characters that matches any of the "Dr.", "Mr.", "Hon.", "Mrs.", "Ms.", "etc.", "i.e.", "e.g.", "[a-zA-Z0-9]+", words ending with apostrophe and all special characters except the currency symbols £ and $. Similarly for the Hindi, a token consisted of a sequence of characters with spaces on both ends and all special characters except the currency symbols £ and $.

3 Word Alignment

Given a pair of parallel sentences, the task of word alignment can be described as finding one-to-one, one-to-many, and many-to-many correspondences

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 115–118,
Ann Arbor, June 2005.

between the words of source and target sentences. It becomes more complicated when aligning phrases of one language with the corresponding words or phrases in the target language. For some words, it is also possible not to find any translation in the target language. Such words are aligned to null.

The algorithm presented in this paper, is a blend of various methods. We categorize words of a Hindi sentence into one of four different categories and use different techniques to deal with each of them. These categories include: 1) NEs and cognates 2) Hindi words for which it is possible to predict their corresponding English words 3) Hindi words that match certain pre-specified regular expression patterns specified in a rule file (explained in section 3.3.) and finally 4) words which do not fit in any of the above categories. In the following sections we explain different methods to deal with words from each of these categories.

3.1 Named Entities and Cognates

According to WWW1, the Named Entity Task is the process of annotating expressions in the text that are "unique identifiers" of entities (e.g. Organization, Person, Location etc.). For example: "Mr. Niraj Aswani", "United Kingdom", and "Microsoft" are examples of NEs. In most text processing systems, this task is achieved by using local pattern-matching techniques e.g. a word that is in upper initial orthography or a Title followed by the two adjacent words that are in upper initial or in all upper case. We use a Hindi gazetteer list that contains a large set of NEs. This gazetteer list is distributed as a part of Hindi Gazetteer processing resource in GATE (Maynard et al., 2003). The Gazetteer list contains various NEs including person names, locations, organizations etc. It also contains other entities such as time units – months, dates, and number expressions. Cognates can be defined as two words having a common etymology and thus are similar or identical. In most cases they are pronounced in a similar way or with a minor change. For example "Bungalow" in English is derived from the word "बंगला" in Hindi, which means a house in the Bengali style (WWW2). We use our TS method to locate such words. Section 3.2 describes the TS approach.

3.2 Transliteration Similarity

For the English-Hindi alphabets, it is possible to come up with a table consisting of correspondences between the letters of the two alphabets. This table is generated based on the various sounds that each letter can produce. For example a letter "c" can be mapped to two letters in Hindi, "क" and "स". This mapping is not restricted to one-to-one but also includes many-to-many correspondences. It is also possible to map a sequence of two or more characters to a single character or to a sequence two or more characters. For example "tio" and "sh" in English correspond to the character "श" in Hindi.

Prior to executing our word alignment algorithm, we use the TS approach to build a table of NEs and cognates. We consider one pair of parallel sentences at a time and for each word in a Hindi sentence, we generate different English words using our TS table. We found that before comparing words of two languages, it is more accurate to eliminate vowels from the words except those that appear at the start of words. We use a dynamic programming algorithm called "edit-distance" to measure the similarity between these words (WWW3). We calculate the similarity measure for each word in a Hindi sentence by comparing it with each and every word of an English sentence. We come up with an m x n matrix, where m and n refer to the number of words in Hindi and English respectively. This matrix contains a similarity measure for each word in a Hindi sentence corresponding to each word in a parallel English sentence. From our experiments of comparing more than 100 NE and cognate pairs, we found that the word pairs should be considered valid matches only if the similarity is greater than 75%. Therefore, we consider only those pairs which have the highest similarity among the other pairs with similarity greater than 75%. The following example shows how TS is used to compare a pair of English-Hindi words. For example consider a pair "aswani → अ।सवानी" and the TS table entries as shown below:

A→अ, S→स, SS→स, V→व, W→व and N→न

We remove vowels from both words: "aswn → असवन", and then convert the Hindi word into possible English words. This gives four different combinations: "asvn", "assvn", "aswn" and "asswn". These words are then compared with the actual English word "aswn". Since we are able to locate at least one word with similarity greater than 75%, we consider "aswani → आसवानी" as a NE. Once a list of NEs and cognates is ready, we switch to our next step: local word grouping, where all words in Hindi sentences, either those available in the gazetteer list or in the list derived using TS approach, are aligned using TS approach.

3.3 Local Word Grouping

Hindi is a partially free order language (i.e. the order of the words in a Hindi sentence is not fixed but the order of words in a group/phrase is fixed). Unlike English where the verbs are used in different inflected forms to indicate different tenses, Hindi uses one or two extra words after the verb to indicate the tense. Therefore, if the English verb is not in its base form, it needs to be aligned with one or more words in a parallel Hindi sentence. Sometimes a phrase is aligned with another phrase. For example "customer benefits" aligns with "ग्राहक के फायदे". In this example the first word "customer" aligns with the first word "ग्राहक" and the second word "benefits" aligns with the third word "फायदे". Considering "customer satisfaction" and "ग्राहक के फायदे" as phrases to be aligned with each other, "के" is the word that indicates the relation between the two words "ग्राहक" and "फायदे", which means the "benefits of customer" in English. These words in a phrase need to be grouped together in order to align them correctly. In the case of certain prepositions, pronouns and auxiliaries, it is possible to predict the respective Hindi postpositions, pronouns and other words. We derived a set of more than 250 rules to group such patterns by consulting the provided training data and other grammar resources such as Bal Anand (2001). The rule file contains the following information for each rule:

1) Hindi Regular Expression for a word or phrase. This must match one or more words in the Hindi sentence.
2) Group name or a part-of-speech category.
3) Expected English word(s) that this Hindi word group may align to.
4) In case a group of one or more English words aligns with a group of one or more Hindi words, information about the key words in both groups. Key words must match each other in order to align English-Hindi groups.
5) A rule to convert Hindi word into its base form.

We list some of the derived rules below:

1) Group a sequence of [X + Postposition], where X can be any category in the above list except postposition or verb. For example: "For X" = "X के लिये", where "For" = "के लिये".
2) Root Verb + (रहा, रही or रहे) + (PH). Present continuous tense. We use "PH" as an abbreviation to refer to the present/past tense conjunction of the verb "होना" - हुं, हैं, है, हो, etc.
3) Group two words that are identical to each other. For example: "अलग अलग", which means "different" in English. Such bi-grams are common in Hindi and are used to stress the importance of a word/activity in a sentence.

Once the words are grouped in a Hindi sentence, we identify those word groups which do not fit in any of the TS and EEW categories. Such words are then aligned using the DL approach.

3.3 Dictionary lookup

Since the most dictionaries contain verbs in their base forms, we use a morphological analyzer to convert verbs in their base forms. The English-Hindi dictionary is obtained from (WWW4). The dictionary returns, on average, two to four Hindi words referring to a particular English word. The formula for finding the lemma of any Hindi verb is: infinitive = root verb + "ना". Since in most cases, our dictionary contains Hindi verbs in their infinitive forms, prior to comparing the word with the unaligned words, we remove the word "ना" from the end of it. Due to minor spelling mistakes it is also possible that the word returned from dictionary does not match with any of the words in

a Hindi sentence. In this case, we use edit-distance algorithm to obtain similarity between the two words. If the similarity is greater than 75%, we consider them similar. We use EEW approach for the words which remain unaligned after the DL approach.

3.4 Expected English words

Candidates for the EEW approach are the Hindi word groups (HWG) that are created by our Hindi local word grouping algorithm (explained in section 3.3). The HWGs such as postpositions, number expressions, month-units, day-units etc. are aligned using the EEW approach. For example, for the Hind word "बावन" in a Hindi sentence, which means "fifty two" in English, the algorithm tries to locate "fifty two" in its parallel English sentence and aligns them if found. For the remaining unaligned Hindi words we use the NAN approach.

3.5 Nearest Aligned Neighbors

In certain cases, words in English-Hindi phrases follow a similar order. The NAN approach works on this principle and aligns one or more words with one of the English words. Considering one HWG at a time, we find the nearest Hindi word that is already aligned with one or more English word(s). Aligning a phrase "customer benefits" with "ग्राहक के फायदे" (example explained in section 3.3) is an example of NAN approach. Similarly consider a phrase "tougher controls", where for its equivalent Hindi phrase "अधिक नियंत्रण", the dictionary returns a correct pair "controls → नियंत्रण", but fails to locate "tougher → अधिक". For aligning the word "tougher", NAN searches for the nearest aligned word, which, in this case, is "controls". Since the word "controls" is already aligned with the word "नियंत्रण", the NAN method aligns the word "tougher" with the nearest unaligned word "अधिक".

4 Test Data results

We executed our algorithm on the test data consisting of 90 English-Hindi sentence pairs. We obtained the following results for non-null alignment pairs.

Word Alignment Evaluation
Evaluation of SURE alignments Precision = 0.7703 Recall = 0.6068 F-measure = 0.6788
Evaluation of PROBABLE alignments Precision = 0.7703 Recall = 0.6068 F-measure = 0.6788
AER = 0.3212

References

Bal Anand, 2001, *Hindi Grammar Books for standard 5 to standard 10*, Navneet Press, India.

Baker P., Bontcheva K., Cunningham H., Gaizauskas R., Hamza O., Hardie A., Jayaram B.D., Leisher M., McEnery A.M., Maynard D., Tablan V., Ursu C., Xiao Z., 2004, *Corpus linguistics and South Asian languages: Corpus creation and tool development,* Literary and Linguistic Computing, 19(4), pp. 509-524.

Maynard D., Tablan V., Bontcheva K., Cunningham H., 2003, *Rapid customisation of an Information Extraction system for surprise languages*, ACM Transactions on Asian Language Information Processing, Special issue on Rapid Development of Language Capabilities: The Surprise Languages.

WWW1, Named Entity Task Definition, http://www.cs.nyu.edu/cs/faculty/grishman/NEtask20.book_2.html#HEADING1 [15/04/2005]

WWW2, Britannica Online Encyclopaedia, http://www.britannica.com/eb/article?tocId=9018081 [15/04/2005]

WWW3, Dynamic Programming Algorithm (DPA) for Edit-Distance, http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/ [22/03/05]

WWW4, English-Hindi dictionary source, http://sanskrit.gde.to/hindi/dict/eng-hin_guj.itx [22/03/05].

Shared Task: Statistical Machine Translation between European Languages

Philipp Koehn
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, UK
pkoehn@inf.ed.ac.uk

Christof Monz
UMIACS
University of Maryland
College Park, MD 20742, USA
christof@umiacs.umd.edu

Abstract

The ACL-2005 Workshop on Parallel Texts hosted a shared task on building statistical machine translation systems for four European language pairs: French–English, German–English, Spanish–English, and Finnish–English. Eleven groups participated in the event. This paper describes the goals, the task definition and resources, as well as results and some analysis.

Statistical machine translation is currently the dominant paradigm in machine translation research. Annual competitions are held for Chinese–English and Arabic–English by NIST (sponsored by the US military funding agency DARPA), which creates a forum to present and compare novel ideas and leads to steady progress in the field.

One of the advantages of statistical machine translation is that the currently applied methods are fairly language-independent. Building a new machine translation system for a new language pair is not much more than a matter of running a training process on a training corpus of parallel text (a text in one language paired with a translation in another).

It is therefore possible to hold a competition where research groups have only a few weeks to build machine translation systems for language pairs that they have not previously worked on. We effectively demonstrated this with our shared task. For instance, seven teams built Finnish–English machine translation systems, a language pair that was certainly not of their immediate concern before.

In contrast to the bigger NIST competition, we wanted to keep the barrier of entry as low as possible. We provided not only training data from the Europarl corpus (Koehn, 2005), but also additional resources: sentence and word alignments, the decoder Pharaoh[1] (Koehn, 2004b), and a language model, so that participation was feasible even as a graduate level class project.

Using about 15 million words of translated text, participants were asked to build a phrase-based statistical machine translation system. The focus of the task was to build a probabilistic phrase translation table, since most of the other resources were provided — for more on phrase-based statistical machine translation, refer to Koehn et al. (2003). The participants' systems were compared by how well they translated 2000 previously unseen test sentences from the same domain.

The shared task operated within an extremely short timeframe. The workshop and hence the shared task was accepted on February 22, 2005 and announced on March 3. The official test data was made available on April 3, results were due one week later. Despite this tight schedule, eleven research groups participated and built a total of 32 machine translation systems for the four language pairs.

1 Goals

When setting up this competition, we were motivated by a number of goals. We set out to:

Create a platform to demonstrate the effectiveness of novel ideas: The research community is easily balkanized, where different groups work on

[1] http://www.isi.edu/licensed-sw/pharaoh/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 119–124,
Ann Arbor, June 2005.

different data sets and under different conditions, so that it becomes often hard to assess, how effective a novel method is. By creating an environment with common test and training sets, language model, preprocessing, and even decoder, the effect of other model choices can be more easily demonstrated.

Work on new language pairs, new problems: Different language pairs pose different challenges. We picked Finnish–English and German–English for the special problems of rich morphology, word order, which are a challenge to current phrase-based SMT methods.

Enable more researchers to get engaged in SMT research: One of our main goals with providing as many resources as possible was to keep the barrier of entry low. Participants could use the word alignment and other resources and focus on phrase extraction. We hoped to attract researchers that are relatively new to the field. We were satisfied to learn that many entries are by graduate students working on their own.

Promote and create free resources: Academic research thrives on freely available resources. The field of statistical machine translation has been blessed with a long tradition of freely available software tools — such as GIZA++ (Och and Ney, 2003) — and parallel corpora — such as the Canadian Hansards[2]. Following this lead, we made word alignments and a language model available for this competition in addition to our previously published resources (Europarl and Pharaoh). The competition created resources as well. Most teams agreed to share system output and their model files. You can download them from the competition web site[3].

Promote work on European language pairs: Finally, we wanted to promote work on European languages. The increasing economic and political ties within the European Union create a huge need for translation services. We would like to see researchers rise to the challenge of creating high quality machine translation systems to fill these needs.

We are very grateful for the strong participation, especially by researchers who are relatively new to the field.

[2]http://www.isi.edu/natural-language/download/hansard/

[3]http://www.statmt.org/wpt05/mt-shared-task/

2 Rules of Engagement

We set up a machine translation competition for four language pairs. We chose Spanish–English and French–English, because many researchers would be familiar with these languages. We chose German–English for its special problems with word order (such as nested constructions and split verb groups) and morphology. Finally, we picked Finnish–English for the rich agglutinative morphology of Finnish.

Statistical machine translation systems are typically trained on sentence-aligned parallel corpora. We selected Europarl[4], a freely available parallel corpus in eleven languages. In addition, we also made a word alignment available, which was derived using a variant of the current default method for word alignment – Och and Ney (2003)'s refined method.

Figure 1 details some properties of the parallel corpora. The training corpus is most of the Europarl corpus, only the text of sessions from last quarter of the year 2000 was reserved for testing. The corpus has the size of roughly 15 million English words in 700,000 sentences – these numbers differ for each of the four parallel corpora due to the different number of discarded sentences during sentence alignment and after enforcing a 40 word length limit for sentences.

The number of foreign words differs even more dramatically. The effect of Finnish morphology manifests itself in a low number of words (just over 11 million), but a high number of distinct words (more than 5 times as many as in the English half).

The test corpus consists of 2000 sentences aligned across all five languages. Note that the output of each system is compared against the same English references for all source languages. The number of total words, distinct words, and words not seen in the training data reflects again the morphology effect.

For researchers willing to create their own word alignment, we suggested the use of GIZA++[5], an implementation of the IBM word-based machine translation models, which also assisted the creation of the provided word alignments.

We trained a language model on the English part

[4]http://www.statmt.org/europarl/

[5]http://www.fjoch.com/GIZA++.html

	Spanish–English	French–English	Finnish–English	German–English
	Training corpus			
Sentences	730,740	688,031	716,960	751,088
Source words	15,676,710	15,323,737	11,318,287	15,256,793
English words	15,222,105	13,808,104	15,492,903	16,052,269
Distinct source words	102,886	80,349	358,345	195,291
Distinct English words	64,123	61,627	64,662	65,889
	Test corpus			
Sentences	2,000			
Source words	60,276	65,029	41,431	54,247
English words	57,945			
Distinct source words	7,782	7,285	11,996	8,666
Distinct English words	6,054			
Unseen source words	209	143	737	377

Figure 1: Properties of the Europarl training and test corpora used in the shared task

of the Europarl corpus using the SRI language modeling toolkit (Stolke, 2002). Finally, we suggested the use of Pharaoh (Koehn, 2004b), a phrase-based machine translation decoder.

How well does this setup match the state of the art? The MIT system using the Pharaoh decoder (Koehn, 2004a) proved to be very competitive in last year's NIST evaluation. However, the field is moving fast, and a number of steps help to improve upon the provided baseline setup, e.g., larger language models trained on general text (up to a billion words have been used), better reodering models (e.g., suggested by Tillman (2004) and Och et al. (2004)), better language-specific preprocessing (Koehn and Knight, 2003) and restructuring (Collins et al., 2005), additional feature functions such as word class language models, and minimum error rate training (Och, 2003) to optimize parameters.

Some of these steps (e.g., improved reordering models) go beyond the current capabilities of Pharaoh. However, we are hopeful that freely available software continues to match or at least follow closely the state of the art.

We announced the shared task on March 3, and provided all the resources mentioned above (also a development test corpus to track the quality of systems being developed). The test schedule called for the translation of 2000 sentence for each of the four language pairs in the week between April 3–10. We allowed late submissions up to April 17.

3 Results

Eleven teams from eight institutions in Europe and North America participated, see Figure 2 for a complete list. The figure also indicates, if a team used the Pharaoh decoder (eight teams), the provided language model (seven teams) and the provided word alignment (four did, three of those with additional preprocessing or additional data).

Translation performance was measured using the BLEU score (Papineni et al., 2002), which measures n-gram overlap with a reference translation. In our case, we only used a single reference translation, since the test set was taken from a held-out portion of the Europarl corpus. On the other hand we used a relatively large number of test sentences to guarantee that the BLEU results are stable despite the fact that we used only one reference translation for each sentence.

Shared tasks like this one, of course, bring out the competitive spirit of participants and can draw criticisms about being a horse race. From an outside perspective, however, it is far more interesting to learn which methods and ideas proved to be successful, than who won the competition.

Taking stock of the results — see Figure 3 — one observes a very packed field at the top. While the participants from the University of Washington produced the best translations for every single language pair, the distance to many other participant scores

ID	Team	Pharaoh	LM	Word Al.
cmu-b	Carnegie Mellon University, USA - Bing Zhao	yes	yes	no
cmu-j	Carnegie Mellon University, USA - Ying (Joy) Zhang	yes	yes	no
glasgow	University of Glasgow, UK	yes	yes	yes+
nrc	National Research Council, Canada	no	no	no
rali	University of Montreal / RALI, Canada	yes	yes	no
saar	Saarland University, Germany	yes	yes	yes
uji	University Jaume I, Spain	yes	yes	yes+
upc-j	Polytechnic University of Catalonia, Spain - Jesus Gimenez	yes	yes	no
upc-m	Polytechnic University of Catalonia, Spain - Marta Ruiz	no	no	no
upc-r	Polytechnic University of Catalonia, Spain - Rafael Banchs	no	no	no
uw	University of Washington, USA	yes	no	yes+

Figure 2: The eleven participating teams: the table also lists, if the Pharaoh decoder, the provided language model, and the provided word alignment was used (yes+ indicates additional preprocessing)

is within a BLEU percentage point or two. As one might have expected, the scores are best for Spanish and French, and worst for Finnish. Figure 4 shows some typical output of the submitted systems.

The proceedings to the workshop include detailed system descriptions of all participants. Novel phrase extraction approaches were proposed, along with better preprocessing, language modeling, rescoring, and other ideas. We are certain that better performance can be achieved by combining some of the methods used by different participants.

And hence, we would like to pose the challenge to the research community to build and test better systems using the provided resources. We will gladly list additional results on the competition web site.

4 Survey

Following the end of the competition, we sent out a questionnaire to the participants. One of the questions what they would like to see different in a potential future competition.

We listed four potential changes: 70% of the respondends checked *translation from English*, 50% checked *out of domain test data*, 40% checked *more language pairs*, 0% checked *fewer language pairs*.

Additional suggestions were: alternatives to the BLEU scoring method (maybe human judgment by participants themselves), transitive translation using pivot languages, translation of resource-poor languages, and more time to prepare for the task.

5 Outlook

Given the short timeframe, one should view the system performances (albeit very competitive with the state of the art) as a baseline effort on the task of open domain text translation between European languages.

We hope that future researchers will use the provided environment as a test bed for their machine translation systems. We will continue to publish any scores reported to us.

Since we placed much of the systems' output online, the interested reader may be inspired to more closely explore the quality and shortcomings. Even some of the model files have been made available, so it is even possible to download and install some of the systems.

References

Collins, M., Koehn, P., and Kucerova, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05), Main Volume*, pages 531–540, Ann Arbor, Michigan.

Koehn, P. (2004a). The foundation for statistical machine translation at MIT. In *Proceedings of Machine Translation Evaluation Workshop 2004*.

Koehn, P. (2004b). Pharaoh: a beam search decoder for statistical machine translation. In *6th Conference of the Association for Machine Translation*

Spanish-English

System	BLEU	1/2/3/4-gram precision (bp)
uw	30.95	64.1/36.6/24.0/16.3 (1.000)
upc-r	30.07	63.1/35.8/23.2/15.6 (1.000)
upc-m	29.84	63.9/35.5/23.0/15.5 (0.995)
nrc	29.08	62.7/34.9/22.2/14.7 (1.000)
rali	28.49	62.4/34.5/21.9/14.4 (0.992)
upc-j	28.13	61.5/33.8/21.4/14.1 (1.000)
saar	26.69	61.0/33.1/20.7/13.5 (0.973)
cmu-j	26.14	61.2/32.4/19.8/12.6 (0.986)
uji	21.65	59.7/27.8/15.2/8.7 (1.000)

French-English

System	BLEU	1/2/3/4-gram precision (bp)
uw	30.27	64.8/36.8/23.8/16.0 (0.981)
upc-r	30.20	63.9/36.2/23.3/15.6 (0.998)
nrc	29.53	63.7/35.8/22.7/14.9 (0.997)
rali	28.89	62.6/34.7/22.0/14.6 (1.000)
cmu-b	27.65	63.1/34.0/20.9/13.3 (0.995)
cmu-j	26.71	61.9/33.0/20.3/13.1 (0.984)
saar	26.29	60.8/32.5/20.1/12.9 (0.982)
glasgow	23.01	57.3/28.0/16.7/10.5 (1.000)
uji	21.25	59.8/27.7/14.8/8.3 (1.000)

Finnish-English

System	BLEU	1/2/3/4-gram precision (bp)
uw	22.01	59.0/28.6/16.1/9.4 (0.979)
nrc	20.95	57.8/27.2/14.8/8.4 (0.996)
upc-r	20.31	56.6/26.0/14.3/8.3 (0.993)
rali	18.87	55.2/24.7/13.1/7.1 (0.998)
saar	16.76	58.4/26.3/14.2/8.0 (0.819)
uji	13.79	60.0/23.2/10.8/5.3 (0.821)
cmu-j	12.66	53.9/21.7/10.7/5.7 (0.775)

German-English

System	BLEU	1/2/3/4-gram precision (bp)
uw	24.77	62.2/31.8/18.8/11.7 (0.965)
upc-r	24.26	59.7/30.1/17.6/11.0 (1.000)
nrc	23.21	60.3/29.8/17.1/10.3 (0.979)
rali	22.91	58.9/29.0/16.8/10.3 (0.982)
saar	20.48	58.0/27.5/15.5/9.2 (0.938)
cmu-j	18.93	59.2/26.8/14.3/8.1 (0.914)
uji	18.89	59.3/25.5/13.0/7.2 (0.976)

Figure 3: The scores for the participating systems (BLEU and its components n-gram-precision and brevity penalty)

in the Americas, AMTA, Lecture Notes in Computer Science. Springer.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit X (submitted)*.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Och, F. J. (2003). Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., and Radev, D. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Stolke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Tillman, C. (2004). A unigram orientation model for statistical machine translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Reference
We know all too well that the present Treaties are inadequate and that the Union will need a better and different structure in future , a more constitutional structure which clearly distinguishes the powers of the Member States and those of the Union .

Input Spanish
Sabemos muy bien que los Tratados actuales no bastan y que , en el futuro , será necesario desarrollar una estructura mejor y diferente para la Unión Europea , una estructura más constitucional que también deje bien claras cuáles son las competencias de los Estados miembros y cuáles pertenecen a la Unión .
Best system (Spanish–English)
we all know very well that the current treaties are not enough and that , in the future , it will be necessary to develop a structure better and different for the european union , a structure more constitutional also make it clear what the competences of the member states and what belongs to the union .
Worst System (Spanish–English)
we know very well that the current treaties not enough and that , in the future , will be necessary develop a better structure and different to the european union , a structure more constitutional that also be well clear the powers of the member states and what belong to the union .

Input French
Nous savons très bien que les Traités actuels ne suffisent pas et qu ' il sera nécessaire à l ' avenir de développer une structure plus efficace et différente pour l ' Union , une structure plus constitutionnelle qui indique clairement quelles sont les compétences des états membres et quelles sont les compétences de l ' Union .
Best system (French–English)
we know very well that the current treaties are not enough and that it will be needed in the future to develop a structure more effective and different for the union , a structure more constitutional which clearly indicates what are the competence of member states and what are the powers of the union .

Input Finnish
Tiedämme oikein hyvin , että nykyiset perustamissopimukset eivät ole riittäviä ja että tulevaisuudessa on tarpeen kehittää unionille parempi ja toisenlainen rakenne , siis perustuslaillisempi rakenne , jossa mys ilmaistaan selkeämmin , mitä jäsenvaltioiden ja unionin toimivaltaan kuuluu
Best system (Finnish–English)
we know very well that the existing founding treaties do not need to be developed for the union and a different structure , therefore perustuslaillisempi structure , which also expresses clearly what the member states and the union 's competence is not sufficient and that better in the future .

Input German
Uns ist sehr wohl bewusst , dass die geltenden Verträge unzulänglich sind und künftig eine andere , effizientere Struktur für die Union entwickelt werden muss , nämlich eine stärker konstitutionell ausgeprägte Struktur mit einer klaren Abgrenzung zwischen den Befugnissen der Mitgliedstaaten und den Kompetenzen der Union .
Best system (German–English)
the union must be developed , with a major institutional structure with a clear demarcation between the powers of the member states and the competences of the union is well aware that the existing treaties are inadequate and in the future , a different , more efficient structure for us .

Figure 4: The first sentence of the test corpus and system translations

Improved Language Modeling for Statistical Machine Translation

Katrin Kirchhoff and Mei Yang
Department of Electrical Engineering
University of Washington, Seattle, WA, 98195
{katrin,yangmei}@ee.washington.edu

Abstract

Statistical machine translation systems use a combination of one or more translation models and a language model. While there is a significant body of research addressing the improvement of translation models, the problem of optimizing language models for a specific translation task has not received much attention. Typically, standard word trigram models are used as an out-of-the-box component in a statistical machine translation system. In this paper we apply language modeling techniques that have proved beneficial in automatic speech recognition to the ACL05 machine translation shared data task and demonstrate improvements over a baseline system with a standard language model.

1 Introduction

Statistical machine translation (SMT) makes use of a noisy channel model where a sentence $\bar{e}$ in the desired language can be conceived of as originating as a sentence $\bar{f}$ in a source language. The goal is to find, for every input utterance $\bar{f}$, the best hypothesis $\bar{e}^*$ such that

$$\bar{e}^* = argmax_{\bar{e}} P(\bar{e}|\bar{f}) = argmax_{\bar{e}} P(\bar{f}|\bar{e})P(\bar{e}) \quad (1)$$

$P(\bar{f}|\bar{e})$ is the translation model expressing probabilistic constraints on the association of source and target strings. $P(\bar{e})$ is a language model specifying the probability of target language strings. Usually, a standard word trigram model of the form

$$P(e_1, ..., e_l) \approx \prod_{i=3}^{l} P(e_i|e_{i-1}, e_{i-2}) \quad (2)$$

is used, where $\bar{e} = e_1, ..., e_l$. Each word is predicted based on a history of two preceding words.

Most work in SMT has concentrated on developing better translation models, decoding algorithms, or minimum error rate training for SMT. Comparatively little effort has been spent on language modeling for machine translation. In other fields, particularly in automatic speech recognition (ASR), there exists a large body of work on statistical language modeling, addressing e.g. the use of word classes, language model adaptation, or alternative probability estimation techniques. The goal of this study was to use some of the language modeling techniques that have proved beneficial for ASR in the past and to investigate whether they transfer to statistical machine translation. In particular, this includes language models that make use of morphological and part-of-speech information, so-called factored language models.

2 Factored Language Models

A factored language model (FLM) (Bilmes and Kirchhoff, 2003) is based on a representation of words as feature vectors and can utilize a variety of additional information sources in addition to words, such as part-of-speech (POS) information, morphological information, or semantic features, in a unified and principled framework. Assuming that each

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 125–128,
Ann Arbor, June 2005.

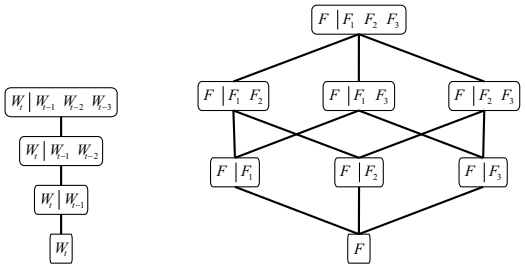
word w can be decomposed into k features, i.e. $w \equiv f^{1:K}$, a trigram model can be defined as

$$p(f_1^{1:K}, f_2^{1:K}, ..., f_T^{1:K}) \approx \prod_{t=3}^{T} p(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K}) \quad (3)$$

Each word is dependent not only on a single stream of temporally preceding words, but also on additional parallel streams of features. This representation can be used to provide more robust probability estimates when a particular word n-gram has not been observed in the training data but its corresponding feature combinations (e.g. stem or tag trigrams) has been observed. FLMs are therefore designed to exploit sparse training data more effectively. However, even when a sufficient amount of training data is available, a language model utilizing morphological and POS information may bias the system towards selecting more fluent translations, by boosting the score of hypotheses with e.g. frequent POS combinations. In FLMs, word feature information is integrated via a new *generalized parallel backoff* technique. In standard Katz-style backoff, the maximum-likelihood estimate of an n-gram with too few observations in the training data is replaced with a probability derived from the lower-order $(n-1)$-gram and a backoff weight as follows:

$$p_{BO}(w_t|w_{t-1}, w_{t-2}) \quad (4)$$
$$= \begin{cases} d_c p_{ML}(w_t|w_{t-1}, w_{t-2}) \text{ if } c > \tau \\ \alpha(w_{t-1}, w_{t-2}) p_{BO}(w_t|w_{t-1}) \text{ otherwise} \end{cases}$$

where c is the count of (w_t, w_{t-1}, w_{t-2}), p_{ML} denotes the maximum-likelihood estimate, τ is a count threshold, d_c is a discounting factor and $\alpha(w_{t-1}, w_{t-2})$ is a normalization factor. During standard backoff, the most distant conditioning variable (in this case w_{t-2}) is dropped first, followed by the second most distant variable etc., until the unigram is reached. This can be visualized as a backoff *path* (Figure 1(a)). If additional conditioning variables are used which do not form a temporal sequence, it is not immediately obvious in which order they should be eliminated. In this case, several backoff paths are possible, which can be summarized in a backoff *graph* (Figure 1(b)). Paths in this graph can be chosen in advance based on linguistic knowledge, or at run-time based on statistical criteria such as counts in the training set. It



(a) (b)

Figure 1: Standard backoff path for a 4-gram language model over words (left) and backoff graph over word features (right).

is also possible to choose multiple paths and combine their probability estimates. This is achieved by replacing the backed-off probability p_{BO} in Equation 2 by a general function g, which can be any non-negative function applied to the counts of the lower-order n-gram. Several different g functions can be chosen, e.g. the mean, weighted mean, product, minimum or maximum of the smoothed probability distributions over all subsets of conditioning factors. In addition to different choices for g, different discounting parameters can be selected at different levels in the backoff graph. One difficulty in training FLMs is the choice of the best combination of conditioning factors, backoff path(s) and smoothing options. Since the space of different combinations is too large to be searched exhaustively, we use a guided search procedure based on Genetic Algorithms (Duh and Kirchhoff, 2004), which optimizes the FLM structure with respect to the desired criterion. In ASR, this is usually the perplexity of the language model on a held-out dataset; here, we use the BLEU scores of the oracle 1-best hypotheses on the development set, as described below. FLMs have previously shown significant improvements in perplexity and word error rate on several ASR tasks (e.g. (Vergyri et al., 2004)).

3 Baseline System

We used a fairly simple baseline system trained using standard tools, i.e. GIZA++ (Och and Ney, 2000) for training word alignments and Pharaoh (Koehn, 2004) for phrase-based decoding. The training data

was that provided on the ACL05 Shared MT task website for 4 different language pairs (translation from Finnish, Spanish, French into English); no additional data was used. Preprocessing consisted of lowercasing the data and filtering out sentences with a length ratio greater than 9. The total number of training sentences and words per language pair ranged between 11.3M words (Finnish-English) and 15.7M words (Spanish-English). The development data consisted of the development sets provided on the website (2000 sentences each). We trained our own word alignments, phrase table, language model, and model combination weights. The language model was a trigram model trained using the SRILM toolkit, with modified Kneser-Ney smoothing and interpolation of higher- and lower-order ngrams. Combination weights were trained using the minimum error weight optimization procedure provided by Pharaoh. We use a two-pass decoding approach: in the first pass, Pharaoh is run in N-best mode to produce N-best lists with 2000 hypotheses per sentence. Seven different component model scores are collected from the outputs, including the distortion model score, the first-pass language model score, word and phrase penalties, and bidirectional phrase and word translation scores, as used in Pharaoh (Koehn, 2004). In the second pass, the N-best lists are rescored with additional language models. The resulting scores are then combined with the above scores in a log-linear fashion. The combination weights are optimized on the development set to maximize the BLEU score. The weighted combined scores are then used to select the final 1-best hypothesis. The individual rescoring steps are described in more detail below.

4 Language Models

We trained two additional language models to be used in the second pass, one word-based 4-gram model, and a factored trigram model. Both were trained on the same training set as the baseline system. The 4-gram model uses modified Kneser-Ney smoothing and interpolation of higher-order and lower-order n-gram probabilities. The potential advantage of this model is that it models n-grams up to length 4; since the BLEU score is a combination of n-gram precision scores up to length 4, the integration of a 4-gram language model might yield better results. Note that this can only be done in a rescoring framework since the first-pass decoder can only use a trigram language model.

For the factored language models, a feature-based word representation was obtained by tagging the text with Rathnaparki's maximum-entropy tagger (Ratnaparkhi, 1996) and by stemming words using the Porter stemmer (Porter, 1980). Thus, the factored language models use two additional features per word. A word history of up to 2 was considered (3-gram FLMs). Rather than optimizing the FLMs on the development set references, they were optimized to achieve a low perplexity on the oracle 1-best hypotheses (the hypotheses with the best individual BLEU scores) from the first decoding pass. This is done to avoid optimizing the model on word combinations that might never be hypothesized by the first-pass decoder, and to bias the model towards achieving a high BLEU score. Since N-best lists differ for different language pairs, a separate FLM was trained for each language pair. While both the 4-gram language model and the FLMs achieved a 8-10% reduction in perplexity on the dev set references compared to the baseline language model, their perplexities on the oracle 1-best hypotheses were not significantly different from that of the baseline model.

5 N-best List Rescoring

For N-best list rescoring, the original seven model scores are combined with the scores of the second-pass language models using the framework of discriminative model combination (Beyerlein, 1998). This approach aims at an optimal (with respect to a given error criterion) integration of different information sources in a log-linear model, whose combination weights are trained discriminatively. This combination technique has been used successfully in ASR, where weights are typically optimized to minimize the empirical word error count on a held-out set. In this case, we use the BLEU score of the N-best hypothesis as an optimization criterion. Optimization is performed using a simplex downhill method known as amoeba search (Nelder and Mead, 1965), which is available as part of the SRILM toolkit.

Language pair	1st pass	oracle
Fi-En	21.8	29.8
Fr-En	28.9	34.4
De-En	23.9	31.0
Es-En	30.8	37.4

Table 1: First-pass (left column) and oracle results (right column) on the dev set (% BLEU).

Language pair	4-gram	FLM	both
Fi-En	22.2	22.2	22.3
Fr-En	30.2	30.2	30.4
De-En	24.6	24.2	24.6
Es-En	31.4	31.0	31.3

Table 2: Second-pass rescoring results (% BLEU) on the dev set for 4-gram LM, 3-gram FLM, and their combination.

6 Results

The results from the first decoding pass on the development set are shown in Table 1. The second column in Table 1 lists the oracle BLEU scores for the N-best lists, i.e. the scores obtained by always selecting the hypothesis known to have the highest individual BLEU score. We see that considerable improvements can in principle be obtained by a better second-pass selection of hypotheses. The language model rescoring results are shown in Table 2, for both types of second-pass language models individually, and for their combination. In both cases we obtain small improvements in BLEU score, with the 4-gram providing larger gains than the 3-gram FLM. Since their combination only yielded negligible additional improvements, only 4-grams were used for processing the final evaluation sets. The evaluation results are shown in Table 3.

Language pair	baseline	4-gram
Fi-En	21.6	22.0
Fr-En	29.3	30.3
De-En	24.2	24.8
Es-En	30.5	31.0

Table 3: Second-pass rescoring results (% BLEU) on the evaluation set.

7 Conclusions

We have demonstrated improvements in BLEU score by utilizing more complex language models in the rescoring pass of a two-pass SMT system. We noticed that FLMs performed worse than word-based 4-gram models. However, only trigram FLM were used in the present experiments; larger improvements might be obtained by 4-gram FLMs. The weights assigned to the second-pass language models during weight optimization were larger than those assigned to the first-pass language model, suggesting that both the word-based model and the FLM provide more useful scores than the baseline language model. Finally, we observed that the overall improvement represents only a small portion of the possible increase in BLEU score as indicated by the oracle results, suggesting that better language models do not have a significant effect on the overall system performance unless the translation model is improved as well.

Acknowledgements

This work was funded by the National Science Foundation, Grant no. IIS-0308297. We are grateful to Philip Koehn for assistance with Pharaoh.

References

P. Beyerlein. 1998. Discriminative model combination. In *Proc. ICASSP*, pages 481–484.

J.A. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of HLT/NAACL*, pages 4–6.

K. Duh and K. Kirchhoff. 2004. Automatic learning of language model structure. In *Proceedings of COLING*.

P. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*.

J.A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Computing Journal*, 7(4):308–313.

F.J. Och and H. Ney. 2000. Giza++: Training of statistical translation models. http://www-i6.informatik.rwth-aachen.de/ och/software/GIZA++.html.

M.F. Porter. 1980. An algorithm for suffi x stripping. *Program*, 14(3):130–137.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings EMNLP*, pages 133–141.

D. Vergyri et al. 2004. Morphology-based language modeling for Arabic speech recognition. In *Proceedings of ICSLP*.

PORTAGE: A Phrase-based Machine Translation System

Fatiha Sadat[+], Howard Johnson[++], Akakpo Agbago[+], George Foster[+], Roland Kuhn[+], Joel Martin[++] and Aaron Tikuisis[*]

[+] NRC Institute for Information Technology
101 St-Jean-Bosco Street
Gatineau, QC K1A 0R6, Canada

[++] NRC Institute for Information Technology
1200 Montreal Road
Ottawa, ON K1A 0R6, Canada

[*] University of Waterloo
200 University Avenue W.,
Waterloo, Ontario, Canada

firstname.lastname@cnrc-nrc.gc.ca aptikuis@uwaterloo.ca

Abstract

This paper describes the participation of the Portage team at NRC Canada in the shared task[1] of ACL 2005 Workshop on Building and Using Parallel Texts. We discuss Portage, a statistical phrase-based machine translation system, and present experimental results on the four language pairs of the shared task. First, we focus on the French-English task using multiple resources and techniques. Then we describe our contribution on the Finnish-English, Spanish-English and German-English language pairs using the provided data for the shared task.

1 Introduction

The rapid growth of the Internet has led to a rapid growth in the need for information exchange among different languages. Machine Translation (MT) and related technologies have become essential to the information flow between speakers of different languages on the Internet. Statistical Machine Translation (SMT), a data-driven approach to producing translation systems, is becoming a practical solution to the longstanding goal of cheap natural language processing.

In this paper, we describe Portage, a statistical phrase-based machine translation system, which we evaluated on all different language pairs that were provided for the shared task. As Portage is a very new system, our main goal in participating in the workshop was to test it out on different language pairs, and to establish baseline performance for the purpose of comparison against other systems and against future improvements. To do this, we used a fairly standard configuration for phrase-based SMT, described in the next section.

Of the language pairs in the shared task, French-English is particularly interesting to us in light of Canada's demographics and policy of official bilingualism. We therefore divided our participation into two parts: one stream for French-English and another for Finnish-, German-, and Spanish-English. For the French-English stream, we tested the use of additional data resources along with hand-coded rules for translating numbers and dates. For the other streams, we used only the provided resources in a purely statistical framework (although we also investigated several automatic methods of coping with Finnish morphology).

The remainder of the paper is organized as follows. Section 2 describes the architecture of the Portage system, including its hand-coded rules for French-English. Experimental results for the four pairs of languages are reported in Section 3. Section 4 concludes and gives pointers to future work.

2 Portage

Portage operates in three main phases: *preprocessing* of raw data into tokens, with translation suggestions for some words or phrases generated by rules; *decoding* to produce one or more translation hypotheses; and error-driven *rescoring* to choose the best final hypothesis. (A fourth *postprocessing* phase was not needed for the shared task.)

[1] http://www.statmt.org/wpt05/mt-shared-task/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 129–132,
Ann Arbor, June 2005.

2.1 Preprocessing

Preprocessing is a necessary first step in order to convert raw texts in both source and target languages into a format suitable for both model training and decoding (Foster et al., 2003). For the supplied *Europarl* corpora, we relied on the existing segmentation and tokenization, except for French, which we manipulated slightly to bring into line with our existing conventions (e.g., converting *l ' an* into *l' an).* For the *Hansard* corpus used to supplement our French-English resources (described in section 3 below), we used our own alignment based on Moore's algorithm (Moore, 2002), segmentation, and tokenization procedures.

Languages with rich morphology are often problematic for statistical machine translation because the available data lacks instances of all possible forms of a word to efficiently train a translation system. In a language like German, new words can be formed by compounding (writing two or more words together without a space or a hyphen in between). Segmentation is a crucial step in preprocessing languages such as German and Finnish texts.

In addition to these simple operations, we also developed a rule-based component to detect numbers and dates in the source text and identify their translation in the target text. This component was developed on the Hansard corpus, and applied to the French-English texts (i.e. Europarl and Hansard), on the development data in both languages, and on the test data.

2.2 Decoding

Decoding is the central phase in SMT, involving a search for the hypotheses *t* that have highest probabilities of being translations of the current source sentence *s* according to a model for *P(t|s)*. Our model for *P(t|s)* is a log-linear combination of four main components: one or more trigram language models, one or more phrase translation models, a distortion model, and a word-length feature. The trigram language model is implemented in the SRILM toolkit (Stolcke, 2002). The phrase-based translation model is similar to the one described in (Koehn, 2004), and relies on symmetrized IBM model 2 word-alignments for phrase pair induction. The distortion model is also very similar to Koehn's, with the exception of a final cost to account for sentence endings.

To set weights on the components of the log-linear model, we implemented Och's algorithm (Och, 2003). This essentially involves generating, in an iterative process, a set of nbest translation hypotheses that are representative of the entire search space for a given set of source sentences. Once this is accomplished, a variant of Powell's algorithm is used to find weights that optimize BLEU score (Papineni et al, 2002) over these hypotheses, compared to reference translations. Unfortunately, our implementation of this algorithm converged only very slowly to a satisfactory final nbest list, so we used two different ad hoc strategies for setting weights: choosing the best values encountered during the iterations of Och's algorithm (French-English), and a grid search (all other languages).

To perform the actual translation, we used our decoder, *Canoe*, which implements a dynamic-programming beam search algorithm based on that of *Pharaoh* (Koehn, 2004). Canoe is input-output compatible with Pharaoh, with the exception of a few extensions such as the ability to decode either backwards or forwards.

2.3 Rescoring

To improve raw output from Canoe, we used a rescoring strategy: have Canoe generate a list of nbest translations rather than just one, then reorder the list using a model trained with Och's method to optimize BLEU score. This is identical to the final pass of the algorithm described in the previous section, except for the use of a more powerful log-linear model than would have been feasible to use inside the decoder. In addition to the four basic features of the initial model, our rescoring model included IBM2 model probabilities in both directions (i.e., P(s|t) and P(t|s)); and an IBM1-based feature designed to detect whether any words in one language seemed to be left without satisfactory translations in the other language. This *missing-word* feature was also applied in both directions.

3 Experiments on the Shared Task

We conducted experiments and evaluations on Portage using the different language pairs of the shared task. The training data was provided for the shared task as follows:

- Training data of 688,031 sentences in French and English. A similarly sized cor-

pus is provided for Finnish, Spanish and German with matched English translations.
- Development test data of 2,000 sentences in the four languages.

In addition to the provided data, a set of 6,056,014 sentences extracted from Hansard corpus, the official record of Canada's parliamentary debates, was used in both French and English languages. This corpus was used to generate both language and translation models for use in decoding and rescoring.

The development test data was split into two parts: The first part that includes 1,000 sentences in each language with reference translations into English served in the optimization of weights for both the decoding and rescoring models. In this study, number of n-best lists was set to 1,000. The second part, which includes 1,000 sentences in each language with reference translations into English, was used in the evaluation of the performance of the translation models.

3.1 Experiments on the French-English Task

Our goal for this language pair was to conduct experiments on Portage for a comparative study exploiting and combining different resources and techniques:

1. Method E is based on the *Europarl* corpus as training data,
2. Method E-H is based on both *Europarl* and *Hansard* corpora as training data,
3. Method E-p is based on the *Europarl* corpus as training data and *parsing numbers and dates* in the preprocessing phase,
4. Method E-H-p is based on both *Europarl* and *Hansard* corpora as training data and *parsing numbers and date* in the preprocessing phase.

Results are shown in Table 1 for the French-English task. The first column of Table 1 indicates the method, the second column gives results for decoding with Canoe only, and the third column for decoding and rescoring with Canoe. For comparison between the four methods, there was an improvement in terms of BLEU scores when using two language models and two translation models generated from Europarl and Hansard corpora; however, parsing numbers and dates had a negative impact on the translation models. The best BLEU score for our participation at the French-English task was 29.53.

Method	Decoding	Decoding+Rescoring
E	27.71	29.22
E-H	28.71	**29.53**
E-p	26.45	28.21
E-H-p	28.29	28.56

Table 1. BLEU scores for the French-English test sentences

A noteworthy feature of these results is that the improvement given by the out-of-domain Hansard corpus was very slight. Although we suspect that somewhat better performance could have been achieved by better weight optimization, this result clearly underscores the importance of matching training and test domains. A related point is that our number and date translation rules actually caused a performance drop due to the fact that they were optimized for typographical conventions prevalent in Hansard, which are quite different from those used in Europarl.

Our best result ranked third in the shared WPT05 French-English task , with a difference of 0.74 in terms of BLEU score from the first ranked participant, and a difference of 0.67 in terms of BLEU score from the second ranked participant.

3.2 Experiments on other Pairs of Languages

The WPT05 workshop provides a good opportunity to achieve our benchmarking goals with corpora that provide challenging difficulties. German and Finnish are languages that make considerable use of compounding. Finnish, in addition, has a particularly complex morphology that is organized on principles that are quite different from any in English. This results in much longer word forms each of which occurs very infrequently.

Our original intent was to propose a number of possible statistical approaches to analyzing and splitting these word forms and improving our results. Since none of these yielded results as good as the baseline, we will continue this work until we understand what is really needed. We also care very much about translating between French and English in Canada and plan to spend a lot of extra effort on difficulties that occur in this case. Translation between Spanish and English is also becoming more important as a result of increased trade within North America but also functions as a good counterpoint for French-English.

Language Pair	Decoding+Rescoring
Finnish-English	20.95
German-English	23.21
Spanish English	29.08

Table 2 BLEU scores for the Finnish-English, German-English and Spanish-English test sentences

To establish our baseline, the only preprocessing we did was lowercasing (using the provided tokenization). Canoe was run without any special settings, although weights for distortion, word penalty, language model, and translation model were optimized using a grid search, as described above. Rescoring was also done, and usually resulted in at least an extra BLEU point.

Our final results are shown in Table 2. Ranks at the shared WPT05 Finnish-, German-, and Spanish-English tasks were assigned as second, third and fourth, with differences of 1.06, 1.87 and 1.56 in terms of BLEU scores, respectively, compared to the first ranked participant.

4 Conclusion

We have reported on our participation in the shared task of the ACL 2005 Workshop on Building and Using Parallel Texts, conducting evaluations of Portage, our statistical machine translation system, on all four language pairs. Our best BLEU scores for the French-, Finnish-, German-, and Spanish-English at this stage were 29.5, 20.95, 23.21 and 29.08, respectively. In total, eleven teams took part at the shared task and most of them submitted results for all pairs of languages. Our results distinguished the NRC team at the third, second, third and fourth ranks with slight differences with the first ranked participants.

A major goal of this work was to evaluate Portage at its first stage of implementation on different pairs of languages. This evaluation has served to identify some problems with our system in the areas of weight optimization and number and date rules. It has also indicated the limits of using out-of-domain corpora, and the difficulty of morphologically complex languages like Finnish.

Current and planned future work includes the exploitation of comparable corpora for statistical machine translation, greater use of morphological knowledge, and better features for nbest rescoring.

References

Andreas Stolcke. 2002. *SRILM - an Extensible Language Modeling Toolkit*. In ICSLP-2002, 901-904.

Franz Josef Och, Hermann Ney. 2000. *Improved Statistical Alignment Models*. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 2000, 440-447.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, Dragomir Radev. 2004. *A Smorgasbord of Features for Statistical Machine Translation*. In Proceeding of the HLT/NAACL 2004, Boston, MA, May 2004.

George Foster, Simona Gandrabur, Philippe Langlais, Pierre Plamondon, Graham Russell and Michel Simard. 2003. *Statistical Machine Translation: Rapid Development with Limited Resources.* In Proceedings of MT Summit IX 2003, New Orleans, September.

Kevin Knight, Ishwar Chander, Matthew Haines, Vasileios Hatzivassiloglou, Eduard Hovy, Masayo Iida, Steve K. Luk, Richard Whitney, and Kenji Yamada. 1995. *Filling Knowledge Gaps in a Broad-Coverage MT System*. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1995.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation.* In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics ACL, Philadelphia, July 2002, pp. 311-318.

Moore, Robert. 2002. *Fast and Accurate Sentence Alignment of Bilingual Corpora.* In Machine Translation: From Research to Real Users (Proceedings of the 5th Conference of the Association for Machine Translation in the Americas, Tiburon, California), Springer-Verlag, Heidelberg, Germany, pp. 135-244.

Och, F. J. and H. Ney. 2002. *Discriminative Training and Maximum Entropy Models for Statistical Machine Translation*. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, pp. 295–302.

Franz Josef Och, 2003. *Minimum Error Rate Training for Statistical Machine Translation*. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, July.

Philipp Koehn. 2002. *Europarl: A multilingual corpusfor evaluation of machine translation*. Ms., University of Southern California.

Philipp Koehn. 2004. *Pharaoh:* a *Beam Search Decoder for Phrase-based Statistical Machine Translation Models.* In Proceedings of the Association for Machine Translation in the Americas AMTA 2004.

Statistical Machine Translation of Euparl Data by using Bilingual N-grams

Rafael E. Banchs **Josep M. Crego** **Adrià de Gispert** **Patrik Lambert** **José B. Mariño**
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya, Barcelona 08034, Spain
{rbanchs,jmcrego,agispert,lambert,canton}@gps.tsc.upc.edu

Abstract

This work discusses translation results for the four Euparl data sets which were made available for the shared task *"Exploiting Parallel Texts for Statistical Machine Translation"*. All results presented were generated by using a statistical machine translation system which implements a log-linear combination of feature functions along with a bilingual n-gram translation model.

1 Introduction

During the last decade, statistical machine translation (SMT) systems have evolved from the original word-based approach (Brown *et al.*, 1993) into phrase-based translation systems (Koehn *et al.*, 2003). Similarly, the noisy channel approach has been expanded to a more general maximum entropy approach in which a log-linear combination of multiple models is implemented (Och and Ney, 2002).

The SMT approach used in this work implements a log-linear combination of feature functions along with a translation model which is based on bilingual n-grams. This translation model was developed by de Gispert and Mariño (2002), and it differs from the well known phrase-based translation model in two basic issues: first, training data is monotonously segmented into bilingual units; and second, the model considers n-gram probabilities instead of relative frequencies. This model is described in section 2.

Translation results from the four source languages made available for the shared task (es: Spanish, fr: French, de: German, and fi: Finnish) into English (en) are presented and discussed.

The paper is structured as follows. Section 2 describes the bilingual n-gram translation model. Section 3 presents a brief overview of the whole SMT procedure. Section 4 presents and discusses the shared task results and other interesting experimentation. Finally, section 5 presents some conclusions and further work.

2 Bilingual N-gram Translation Model

As already mentioned, the translation model used here is based on bilingual n-grams. It actually constitutes a language model of bilingual units which are referred to as tuples (de Gispert and Mariño, 2002). This model approximates the joint probability between source and target languages by using 3-grams as it is described in the following equation:

$$p(T,S) \approx \prod_{n=1}^{N} p((t,s)_n|(t,s)_{n-2},(t,s)_{n-1}) \quad (1)$$

where t refers to target, s to source and $(t,s)_n$ to the n^{th} tuple of a given bilingual sentence pair.

Tuples are extracted from a word-to-word aligned corpus according to the following two constraints: first, tuple extraction should produce a monotonic segmentation of bilingual sentence pairs; and second, the produced segmentation is maximal in the sense that no smaller tuples can be extracted without violating the previous constraint (Crego *et al.*, 2004). According to this, tuple extraction provides a unique segmentation for a given bilingual sentence pair alignment. Figure 1 illustrates this idea with a simple example.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 133–136,
Ann Arbor, June 2005.

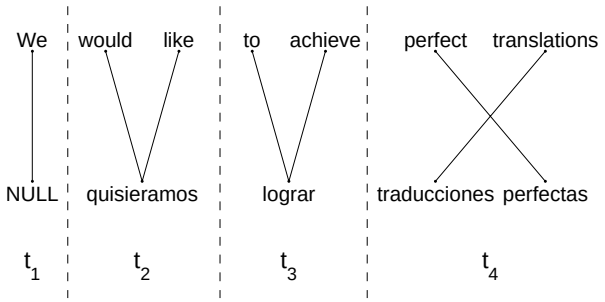


Figure 1: *Example of tuple extraction from an aligned sentence pair.*

Two important issues regarding this translation model must be mentioned. First, when extracting tuples, some words always appear embedded into tuples containing two or more words, so no translation probability for an independent occurrence of such words exists. To overcome this problem, the tuple 3-gram model is enhanced by incorporating 1-gram translation probabilities for all the embedded words (de Gispert *et al.*, 2004).

Second, some words linked to NULL end up producing tuples with NULL source sides. This cannot be allowed since no NULL is expected to occur in a translation input. This problem is solved by preprocessing alignments before tuple extraction such that any target word that is linked to NULL is attached to either its precedent or its following word.

3 SMT Procedure Description

This section describes the procedure followed for preprocessing the data, training the models and optimizing the translation system parameters.

3.1 Preprocessing and Alignment

The Euparl data provided for this shared task (Europarl, 2003) was preprocessed for eliminating all sentence pairs with a word ratio larger than 2.4. As a result of this preprocessing, the number of sentences in each training set was slightly reduced. However, no significant reduction was produced.

In the case of French, a re-tokenizing procedure was performed in which all apostrophes appearing alone were attached to their corresponding words. For example, pairs of tokens such as *l '* and *qu '* were reduced to single tokens such as *l'* and *qu'*.

Once the training data was preprocessed, a word-to-word alignment was performed in both directions, source-to-target and target-to-source, by using GIZA++ (Och and Ney, 2000). As an approximation to the most probable alignment, the Viterbi alignment was considered. Then, the intersection and union of alignment sets in both directions were computed for each training set.

3.2 Feature Function Computation

The considered translation system implements a total of five feature functions. The first of these models is the tuple 3-gram model, which was already described in section 2. Tuples for the translation model were extracted from the union set of alignments as shown in Figure 1. Once tuples had been extracted, the tuple vocabulary was pruned by using histogram pruning. The same pruning parameter, which was actually estimated for Spanish-English, was used for the other three language pairs. After pruning, the tuple 3-gram model was trained by using the SRI Language Modeling toolkit (Stolcke, 2002). Finally, the obtained model was enhanced by incorporating 1-gram probabilities for the embedded word tuples, which were extracted from the intersection set of alignments.

Table 1 presents the total number of running words, distinct tokens and tuples, for each of the four training data sets.

Table 1: *Total number of running words, distinct tokens and tuples in training.*

source language	running words	distinct tokens	tuple vocabulary
Spanish	15670801	113570	1288770
French	14844465	78408	1173424
German	15207550	204949	1391425
Finnish	11228947	389223	1496417

The second feature function considered was a target language model. This feature actually consisted of a word 3-gram model, which was trained from the target side of the bilingual corpus by using the SRI Language Modeling toolkit.

The third feature function was given by a word penalty model. This function introduces a sentence length penalization in order to compensate the sys-

tem preference for short output sentences. More specifically, the penalization factor was given by the total number of words contained in the translation hypothesis.

Finally, the fourth and fifth feature functions corresponded to two lexicon models based on IBM Model 1 lexical parameters $p(t|s)$ (Brown *et al.*, 1993). These lexicon models were calculated for each tuple according to the following equation:

$$p_{lexicon}((t,s)_n) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(t_n^i | s_n^j) \quad (2)$$

where s_n^j and t_n^i are the j^{th} and i^{th} words in the source and target sides of tuple $(t,s)_n$, being J and I the corresponding total number words in each side of it.

The forward lexicon model uses IBM Model 1 parameters obtained from source-to-target alignments, while the backward lexicon model uses parameters obtained from target-to-source alignments.

3.3 Decoding and Optimization

The search engine for this translation system was developed by Crego *et al.* (2005). It implements a beam-search strategy based on dynamic programming and takes into account all the five feature functions described above simultaneously. It also allows for three different pruning methods: threshold pruning, histogram pruning, and hypothesis recombination. For all the results presented in this work the decoder's monotonic search modality was used.

An optimization tool, which is based on a simplex method (Press *et al.*, 2002), was developed and used for computing log-linear weights for each of the feature functions described above. This algorithm adjusts the log-linear weights so that *BLEU* (Papineni *et al.*, 2002) is maximized over a given development set. One optimization for each language pair was performed by using the 2000-sentence development sets made available for the shared task.

4 Shared Task Results

Table 2 presents the *BLEU* scores obtained for the shared task test data. Each test set consisted of 2000 sentences. The computed *BLEU* scores were case insensitive and used one translation reference.

Table 2: *BLEU scores (shared task test sets).*

es - en	fr - en	de - en	fi - en
0.3007	0.3020	0.2426	0.2031

As can be seen from Table 2 the best ranked translations were those obtained for French, followed by Spanish, German and Finnish. A big difference is observed between the best and the worst results.

Differences can be observed from translation outputs too. Consider, for example, the following segments taken from one of the test sentences:

es-en: *We know very well that the present Treaties are not enough and that , in the future , it will be necessary to develop a structure better and different for the European Union...*

fr-en: *We know very well that the Treaties in their current are not enough and that it will be necessary for the future to develop a structure more effective and different for the Union...*

de-en: *We very much aware that the relevant treaties are inadequate and , in future to another , more efficient structure for the European Union that must be developed...*

fi-en: *We know full well that the current Treaties are not sufficient and that , in the future , it is necessary to develop the Union better and a different structure...*

It is evident from these translation outputs that translation quality decreases when moving from Spanish and French to German and Finnish. A detailed observation of translation outputs reveals that there are basically two problems related to this degradation in quality. The first has to do with reordering, which seems to be affecting Finnish and, specially, German translations.

The second problem has to do with vocabulary. It is well known that large vocabularies produce data sparseness problems (Koehn, 2002). As can be confirmed from Tables 1 and 2, translation quality decreases as vocabulary size increases. However, it is not clear yet, in which degree such degradation is due to monotonic decoding and/or vocabulary size.

Finally, we also evaluated how much the full feature function system differs from the baseline tuple 3-gram model alone. In this way, *BLEU* scores were computed for translation outputs obtained for the baseline system and the full system. Since the English reference for the test set was not available, we computed translations and *BLEU* scores over de-

velopment sets. Table 3 presents the results for both the full system and the baseline.[1]

Table 3: *Baseline- and full-system BLEU scores (computed over development sets).*

language pair	baseline	full
es - en	0.2588	0.3004
fr - en	0.2547	0.2938
de - en	0.1844	0.2350
fi - en	0.1526	0.1989

From Table 3, it is evident that the four additional feature functions produce important improvements in translation quality.

5 Conclusions and Further Work

As can be concluded from the presented results, performance of the translation system used is much better for French and Spanish than for German and Finnish. As some results suggest, reordering and vocabulary size are the most important problems related to the low translation quality achieved for German and Finnish.

It is also evident that the bilingual n-gram model used requires the additional feature functions to produce better translations. However, more experimentation is required in order to fully understand each individual feature's influence on the overall log-linear model performance.

6 Acknowledgments

This work has been funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation -(IST-2002-FP6-506738, `http://www.tc-star.org`).

The authors also want to thank José A. R. Fonollosa and Marta Ruiz Costa-jussà for their participation in discussions related to this work.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. "The mathematics of statistical machine translation: parameter estimation". *Computational Linguistics*, 19(2):263–311.

Josep M. Crego, José B. Mariño, and Adrià de Gispert. 2004. "Finite-state-based and phrase-based statistical machine translation". *Proc. of the 8th Int. Conf. on Spoken Language Processing*, :37–40, October.

Josep M. Crego, José B. Mariño, and Adrià de Gispert. 2005. "A Ngram-based Statistical Machine Translation Decoder". Submitted to INTERSPEECH 2005.

Adrià de Gispert, and José B. Mariño. 2002. "Using X-grams for speech-to-speech translation". *Proc. of the 7th Int. Conf. on Spoken Language Processing*.

Adrià de Gispert, José B. Mariño, and Josep M. Crego. 2004. "TALP: Xgram-based spoken language translation system". *Proc. of the Int. Workshop on Spoken Language Translation*, :85–90. Kyoto, Japan, October.

EUPARL: European Parliament Proceedings Parallel Corpus 1996-2003. Available on-line at: `http://people.csail.mit.edu/people/koehn/publications/europarl/`

Philipp Koehn. 2002. "Europarl: A Multilingual Corpus for Evaluation of Machine Translation". Available on-line at: `http://people.csail.mit.edu/people/koehn/publications/europarl/`

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. "Statistical phrase-based translation". *Proc. of the 2003 Meeting of the North American chapter of the ACL*, Edmonton, Alberta.

Franz J. Och and Hermann Ney. 2000. "Improved statistical alignment models". *Proc. of the 38th Ann. Meeting of the ACL*, Hong Kong, China, October.

Franz J. Och and Hermann Ney. 2002. "Discriminative training and maximum entropy models for statistical machine translation". *Proc. of the 40th Ann. Meeting of the ACL*, :295–302, Philadelphia, PA, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. "Bleu: a method for automatic evaluation of machine translation". *Proc. of the 40th Ann. Conf. of the ACL*, Philadelphia, PA, July.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2002. *Numerical Recipes in C++: the Art of Scientific Computing*, Cambridge University Press.

Andreas Stolcke. 2002. "SRLIM: an extensible language modeling toolkit". *Proc. of the Int. Conf. on Spoken Language Processing* :901–904, Denver, CO, September. Available on line at: `http://www.speech.sri.com/projects/srilm/`

[1] Differently from *BLEU* scores presented in Table 2, which are case insensitive, *BLEU* scores presented in Table 3 are case sensitive.

RALI: SMT shared task system description

Philippe Langlais, Guihong Cao and Fabrizio Gotti
RALI
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
Succursale Centre-Ville
H3C3J7 Montréal, Canada
http://rali.iro.umontreal.ca

Abstract

Thanks to the profusion of freely available tools, it recently became fairly easy to built a statistical machine translation (SMT) engine given a bitext. The expectations we can have on the quality of such a system may however greatly vary from one pair of languages to another. We report on our experiments in building phrase-based translation engines for the four pairs of languages we had to consider for the SMT shared-task.

1 Introduction

Machine translation is nowadays mature enough that it is possible without too much effort to devise automatically a statistical translation system from just a parallel corpus. This is possible thanks to the dissemination of valuable packages. The performance of such a system may however greatly vary from one pair of languages to another. Indeed, there is no free lunch for system developers, and if a black box approach can sometimes be good enough for some applications (we can surely accomplish translation *gisting* with the French-English and Spanish-English systems we developed during this exercice), making use of the output of such a system for, let's say, quality translation is another kettle of fish (especially in our case with the Finnish-English system we ended-up with).

We devoted two weeks to the SMT shared task, the aim of which was precisely to see how well systems can do across different language families. We began with a core system which is described in the next section and from which we obtained baseline performances that we tried to improve upon.

Since the French- and Spanish-English systems produced output that were comprehensible enough[1], we focussed on the two languages whose translations were noticeably worse: German and Finnish. For German, we tried to move around words in order to mimic English word order; and we tried to split compound words. This is described in section 4. For the Finnish/English pair, we tried to decompose Finnish words into smaller substrings (see section 5).

In parallel to that, we tried to smooth a phrase-based model (PBM) making use of WORDNET. We report on this experiment in section 3. We describe in section 6 the final setting of the systems we used for submitting translations and their official results as computed by the organizers. Finally, we conclude our two weeks of efforts in section 7.

2 The core system

We assembled up a phrase-based statistical engine by making use of freely available packages. The translation engine we used is the one suggested within the shared task: PHARAOH (Koehn, 2004). The input of this decoder is composed of a phrase-based model (PBM), a trigram language model and an optional set of coefficients and thresholds

[1] What we mean by this is nothing more than we were mostly able to infer the original meaning of the source sentence by reading its automatic translation.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 137–140,
Ann Arbor, June 2005.

pair	WER	SER	NIST	BLEU
fi-en	66.53	99.20	5.3353	18.73
de-en	60.70	98.40	5.8411	21.11
fr-en	53.77	98.20	6.4717	27.69
es-en	53.84	98.60	6.5571	28.08

Table 1: Baseline performances measured on the 500 top sentences of the DEV corpus in terms of WER (word error rate), SER (sentence error rate), NIST and BLEU scores.

which control the decoder.

For acquiring a PBM, we followed the approach described by Koehn et al. (2003). In brief, we relied on a bi-directional word alignment of the training corpus to acquire the parameters of the model. We used the word alignment produced by Giza (Och and Ney, 2000) out of an IBM model 2. We did try to use the alignment produced with IBM model 4, but did not notice significant differences over our experiments; an observation consistent with the findings of Koehn et al. (2003). Each parameter in a PBM can be scored in several ways. We considered its relative frequency as well as its IBM-model 1 score (where the transfer probabilities were taken from an IBM model 2 transfer table). The language model we used was the one provided within the shared task.

We obtained baseline performances by tuning the engine on the top 500 sentences of the development corpus. Since we only had a few parameters to tune, we did it by sampling the parameter space uniformly. The best performance we obtained, *i.e.*, the one which maximizes the BLEU metric as measured by the `mteval` script[2] is reported for each pair of languages in Table 1.

3 Smoothing PBMs with WORDNET

Among the things we tried but which did not work well, we investigated whether smoothing the transfer table of an IBM model (2 in our case) with WORDNET would produce better estimates for rare words. We adapted an approach proposed by Cao et al. (2005) for an Information Retrieval task, and computed for any parameter (e_i, f_j) belonging to the original model the following approximation:

$$\dot{p}(e_i|f_j) \approx \sum_{e \in \mathcal{E}} p_{wn}(e_i|e) \times p_n(e|f_j)$$

where $\mathcal{E}$ is the English vocabulary, p_n designates the native distribution and p_{wn} is the probability that two words in the English side are linked together. We estimated this distribution by co-occurrence counts over a large English corpus[3]. To avoid taking into account unrelated but co-occurring words, we used WORDNET to filter in only the co-occurrences of words that are in relation according to WORDNET. However, since many words are not listed in this resource, we had to smooth the bigram distribution, which we did by applying Katz smoothing (Katz, 1997):

$$p_{katz}(e_i|e) = \begin{cases} \frac{\dot{c}(e_i, e|W,L)}{\sum_{e_j} c(e_j, e|W,L)} & \text{if } c(e_i, e|W, L) > 0 \\ \alpha(e) p_{katz}(e_i) & \text{otherwise} \end{cases}$$

where $\dot{c}(a, b|W, L)$ is the good-turing discounted count of times two words a and b that are linked together by a WORDNET relation, co-occur in a window of 2 sentences.

We used this smoothed model to score the parameters of our PBM instead of the native transfer table. The results were however disappointing for both the G-E and S-E translation directions we tested. One reason for that, may be that the English corpus we used for computing the co-occurrence counts is an out-of-domain corpus for the present task. Another possible explanation lies in the fact that we considered both synonymic and hyperonymic links in WORDNET; the latter kind of links potentially introducing too much noise for a translation task.

4 The German-English task

We identified two major problems with our approach when faced with this pair of languages. First, the tendency in German to put verbs at the end of a phrase happens to ruin our phrase acquisition process, which basically collects any box of aligned source and target adjacent words. This

[2] `http://www.nist.gov/speech/tests/mt/mt2001/resource`

[3] For this, we used the English side of the provided training corpus plus the English side of our in-house Hansard bitext; that is, a total of more than 7 million pairs of sentences.

can be clearly seen in the alignment matrix of figure 1 where the verbal construction *could clarify* is translated by two very distant German words *könnten* and *erläutern*. Second, there are many compound words in German that greatly dilute the various counts embedded in the PBM table.

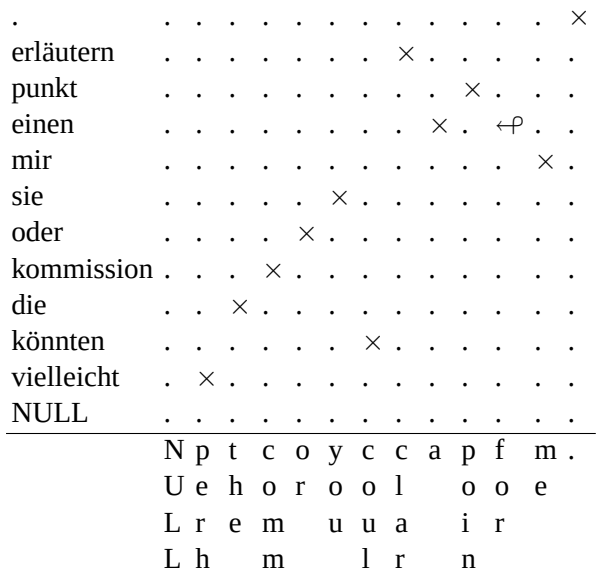


Figure 1: Bidirectional alignment matrix. A cross in this matrix designates an alignment valid in both directions, while the $\leftarrow\!\!\rho$ symbol indicates an uni-directional alignment (`for` has been aligned with `einen`, but not the other way round).

4.1 Moving around German words

For the first problem, we applied a memory-based approach to move around words in the German side in order to better synchronize word order in both languages. This involves, first, to learning transformation rules from the training corpus, second, transforming the German side of this corpus; then training a new translation model. The same set of rules is then applied to the German text to be translated.

The transformation rules we learned concern a few (five in our case) verbal constructions that we expressed with regular expressions built on POS tags in the English side. Once the *locus* e_u^v of a pattern has been identified, a rule is collected whenever the following conditions apply: for each word e in the locus, there is a target word f which is aligned to e in both alignment directions; these target words when moved can lead to a diagonal going from the target word (l) associated to e_{u-1} to the target word r which is aligned to e_{v+1}.

The rules we memorize are triplets (c, i, o) where $c = (l, r)$ is the context of the locus and i and o are the input and output German word order (that is, the order in which the tokens are found, and the order in which they should be moved).

For instance, in the example of Figure 1, the `Verb Verb` pattern match the locus *could clarify* and the following rule is acquired: `(sie einen, könnten erläutern, könnten erläutern)`, a paraphrase of which is: "whenever you find (in this order) the word *könnten* and *erläutern* in a German sentence containing also (in this order) *sie* and *einen*, move *könnten* and *erläutern* between *sie* and *einen*.

A set of 124 271 rules have been acquired this way from the training corpus (for a total of 157 970 occurrences). The most frequent rule acquired is `(ich herrn, möchte danken, möchte danken)`, which will transform a sentence like "*ich möchte herrn wynn für seinen bericht danken.*" into "*ich möchte danken herrn wynn für seinen bericht.*".

In practice, since this acquisition process does not involve any generalization step, only a few rules learnt really fire when applied to the test material. Also, we devised a fairly conservative way of applying the rules, which means that in practice, only 3.5% of the sentences of the test corpus where actually modified.

The performance of this procedure as measured on the development set is reported in Table 2. As simple as it is, this procedure yields a relative gain of 7% in BLEU. Given the crudeness of our approach, we consider this as an encouraging improvement.

4.2 Compound splitting

For the second problem, we segmented German words before training the translation models. Empirical methods for compound splitting applied to

system	WER	SER	NIST	BLEU
baseline	60.70	98.40	5.8411	21.11
swap	60.73	98.60	5.9643	22.58
split	60.67	98.60	5.7511	21.99
swap+split	60.57	98.40	5.9685	23.10

Table 2: Performances of the swapping and the compound splitting approaches on the top 500 sentences of the development set.

German have been studied by Koehn and Knight (2003). They found that a simple splitting strategy based on the frequency of German words was the most efficient method of the ones they tested, when embedded in a phrase-based translation engine. Therefore, we applied such a strategy to split German words in our corpora. The results of this approach are shown in Table 2.

Note: Both the swapping strategy and the compound splitting yielded improvements in terms of BLEU score. Only after the deadline did we find time to train new models with a combination of both techniques; the results of which are reported in the last line of Table 2.

5 The Finnish-English task

The worst performances were registered on the Finnish-English pair. This is due to the agglutinative nature of Finnish. We tried to segment the Finnish material into smaller units (substrings) by making use of the frequency of all Finnish substrings found in the training corpus. We maintained a suffix tree structure for that purpose. We proceeded by recursively finding the most promising splitting points in each Finnish token of C characters F_1^C by computing $split(F_1^C)$ where:

$$split(F_i^j) = \begin{cases} |F_i^j| & \text{if } j-i<2 \\ \max_{c\in[i+2,j-2]} |F_i^c| \times & \\ \quad split(F_{c+1}^j) & \text{otherwise} \end{cases}$$

This approach yielded a significant degradation in performance that we still have to analyze.

6 Submitted translations

At the time of the deadline, the best translations we had were the baselines ones for all the language pairs, except for the German-English one where the moving of words ranked the best. This defined the configuration we submitted, whose results (as provided by the organizers) are reported in Table 3.

pair	BLEU	$p_1/p_2/p_3/p_4$
fi-en	18.87	55.2/24.7/13.1/7.1
de-en	22.91	58.9/29.0/16.8/10.3
es-en	28.49	62.4/34.5/21.9/14.4
fr-en	28.89	62.6/34.7/22.0/14.6

Table 3: Results measured by the organizers for the TEST corpus.

7 Conclusion

We found that, while comprehensible translations were produced for pairs of languages such as French-English and Spanish-English; things did not go as well for the German-English pair and especially not for the Finnish-English pair. We had a hard time improving our baseline performance in such a tight schedule and only managed to improve our German-English system. We were less lucky with other attempts we implemented, among them, the smoothing of a transfer table with WORDNET, and the segmentation of the Finnish corpus into smaller units.

References

G. Cao, J. Nie, and J. Bai. 2005. Integrating Word relationships into Language Models. In *to appear in Proc. of SIGIR*.

S. Katz. 1997. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing*, 35.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL*, Budapest, Hungary.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT*, pages 127–133.

P. Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based SMT. In *Proceedings of AMTA*, pages 115–124.

F.J. Och and H. Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL*, pages 440–447, Hongkong, China.

A Generalized Alignment-Free Phrase Extraction

Bing Zhao
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA-15213
bzhao@cs.cmu.edu

Stephan Vogel
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA-15213
vogel+@cs.cmu.edu

Abstract

In this paper, we present a phrase extraction algorithm using a translation lexicon, a fertility model, and a simple distortion model. Except these models, we do not need explicit word alignments for phrase extraction. For each phrase pair (a block), a bilingual lexicon based score is computed to estimate the translation quality between the source and target phrase pairs; a fertility score is computed to estimate how good the lengths are matched between phrase pairs; a center distortion score is computed to estimate the relative position divergence between the phrase pairs. We presented the results and our experience in the shared tasks on French-English.

1 Introduction

Phrase extraction becomes a key component in today's state-of-the-art statistical machine translation systems. With a longer context than unigram, phrase translation models have flexibilities of modelling local word-reordering, and are less sensitive to the errors made from preprocessing steps including word segmentations and tokenization. However, most of the phrase extraction algorithms rely on good word alignments. A widely practiced approach explained in details in (Koehn, 2004), (Och and Ney, 2003) and (Tillmann, 2003) is to get word alignments from two directions: source to target and target to source; the intersection or union operation is applied to get refined word alignment with pre-designed heuristics fixing the unaligned words. With this refined word alignment, the phrase extraction for a given source phrase is essentially to extract the target candidate phrases in the target sentence by searching the left and right projected boundaries.

In (Vogel et al., 2004), they treat phrase alignment as a sentence splitting problem: given a source phrase, find the boundaries of the target phrase such that the overall sentence alignment lexicon probability is optimal. We generalize it in various ways, esp. by using a fertility model to get a better estimation of phrase lengths, and a phrase level distortion model.

In our proposed algorithm, we do not need explicit word alignment for phrase extraction. Thereby it avoids the burden of testing and comparing different heuristics especially for some language specific ones. On the other hand, the algorithm has such flexibilities that one can incorporate word alignment and heuristics in several possible stages within this proposed framework to further improve the quality of phrase pairs. In this way, our proposed algorithm is more generalized than the usual word alignment based phrase extraction algorithms.

The paper is structured as follows: in section 2, The concept of blocks is explained; in section 3, a dynamic programming approach is model the width of the block; in section 4, a simple center distortion of the block; in section 5, the lexicon model; the complete algorithm is in section 6; in section 7, our experience and results using the proposed approach.

2 Blocks

We consider each phrase pair as a block within a given parallel sentence pair, as shown in Figure 1.

The y-axis is the source sentence, indexed word by word from bottom to top; the x-axis is the target sentence, indexed word by word from left to right. The block is defined by the source phrase and its projection. The source phrase is bounded by the *start* and the *end* positions in the source sentence. The projection of the source phrase is defined as the left and right boundaries in the target sentence. Usually, the boundaries can be inferred according to word alignment as the left most and right most aligned positions from the words in the source phrase. In

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 141–144,
Ann Arbor, June 2005.

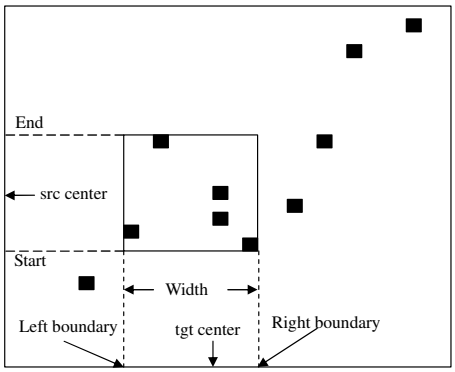


Figure 1: Blocks with "width" and "centers"

this paper, we provide another view of the block, which is defined by the *centers* of source and target phrases, and the *width* of the target phrase.

Phrase extraction algorithms in general search for the left and right projected boundaries of each source phrase according to some score metric computed for the given parallel sentence pairs. We present here three models: a phrase level fertility model score for phrase pairs' length mismatch, a simple center-based distortion model score for the divergence of phrase pairs' relative positions, and a phrase level translation score to approximate the phrase pairs' translational equivalence. Given a source phrase, we can search for the best possible block with the highest combined scores from the three models.

3 Length Model: Dynamic Programming

Given the word fertility definitions in IBM Models (Brown et al., 1993), we can compute a probability to predict *phrase length*: given the candidate target phrase (English) e_1^I, and a source phrase (French) of length J, the model gives the estimation of $P(J|e_1^I)$ via a dynamic programming algorithm using the source word fertilities. Figure 2 shows an example fertility trellis of an English trigram. Each edge between two nodes represents one English word e_i. The arc between two nodes represents one candidate non-zero fertility for e_i. The fertility of zero (i.e. generating a NULL word) corresponds to the direct edge between two nodes, and in this way, the NULL word is naturally incorporated into this model's representation. Each arc is

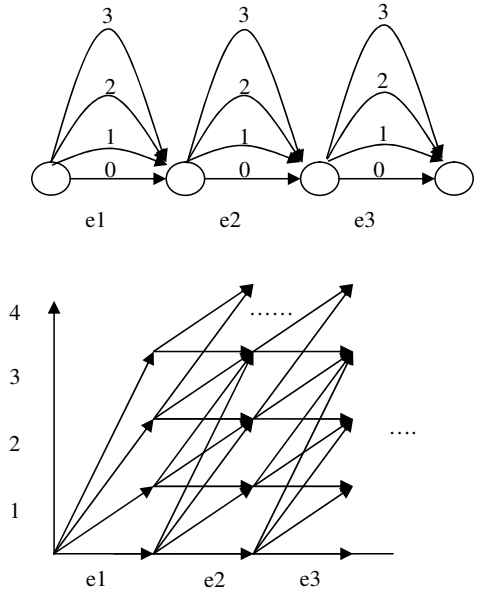


Figure 2: An example of fertility trellis for dynamic programming

associated with a English word fertility probability $P(\phi_i|e_i)$. A path ϕ_1^I through the trellis represents the number of French words ϕ_i generated by each English word e_i. Thus, the probability of generating J words from the English phrase along the Viterbi path is:

$$P(J|e_1^I) = \max_{\{\phi_1^I, J=\sum_{i=1}^I \phi_i\}} \prod_{i=1}^{I} P(\phi_i|e_i) \quad (1)$$

The Viterbi path is inferred via dynamic programming in the trellis of the lower panel in Figure 2:

$$\phi[j,i] = max \begin{cases} \phi[j,i-1] + \log P_{NULL}(0|e_i) \\ \phi[j-1,i-1] + \log P_\phi(1|e_i) \\ \phi[j-2,i-1] + \log P_\phi(2|e_i) \\ \phi[j-3,i-1] + \log P_\phi(3|e_i) \end{cases}$$

where $P_{NULL}(0|e_i)$ is the probability of generating a NULL word from e_i; $P_\phi(k=1|e_i)$ is the usual word fertility model of generating one French word from the word e_i; $\phi[j,i]$ is the cost so far for generating j words from i English words $e_1^i : e_1, \cdots, e_i$.

After computing the cost of $\phi[J,I]$, we can trace back the Viterbi path, along which the probability $P(J|e_1^I)$ of generating J French words from the English phrase e_1^I as shown in Eqn. 1.

With this phrase length model, for every candidate block, we can compute a phrase level fertility score to estimate to how good the phrase pairs are match in their lengthes.

4 Distortion of Centers

The centers of source and target phrases are both illustrated in Figure 1. We compute a simple distortion score to estimate how far away the two centers are in a parallel sentence pair in a sense the block is close to the diagonal.

In our algorithm, the source center $\odot_{f_j^{j+l}}$ of the phrase f_j^{j+l} with length $l+1$ is simply a normalized relative position defined as follows:

$$\odot_{f_j^{j+l}} = \frac{1}{|F|} \sum_{j'=j}^{j'=j+l} \frac{j'}{l+1} \tag{2}$$

where $|F|$ is the French sentence length.

For the center of English phrase e_i^{i+k} in the target sentence, we first define the expected corresponding relative center for every French word $f_{j'}$ using the lexicalized position score as follows:

$$\odot_{e_i^{i+k}}(f_{j'}) = \frac{1}{|E|} \cdot \frac{\sum_{i'=i}^{(i+k)} i' \cdot P(f_{j'}|e_{i'})}{\sum_{i'=i}^{(i+k)} P(f_{j'}|e_{i'})} \tag{3}$$

where $|E|$ is the English sentence length. $P(f_{j'}|e_i)$ is the word translation lexicon estimated in IBM Models. i is the position index, which is weighted by the word level translation probabilities; the term of $\sum_{i=1}^{I} P(f_{j'}|e_i)$ provides a normalization so that the expected center is within the range of target sentence length. The expected center for e_i^{i+k} is simply a average of $\odot_{e_i^{i+k}}(f_{j'})$:

$$\odot_{e_i^{i+k}} = \frac{1}{l+1} \sum_{j'=j}^{j+l} \odot_{e_i^{i+k}}(f_{j'}) \tag{4}$$

This is a general framework, and one can certainly plug in other kinds of score schemes or even word alignments to get better estimations.

Given the estimated centers of $\odot_{f_j^{j+l}}$ and $\odot_{e_i^{i+k}}$, we can compute how close they are by the probability of $P(\odot_{e_i^{i+k}}|\odot_{f_j^{j+l}})$. To estimate $P(\odot_{e_i^{i+k}}|\odot_{f_j^{j+l}})$, one can start with a flat gaussian model to enforce the point of $(\odot_{e_i^{i+k}}, \odot_{f_j^{j+l}})$ not too far off the diagonal and build an initial list of phrase pairs, and then compute the histogram to approximate $P(\odot_{e_i^{i+k}}|\odot_{f_j^{j+l}})$.

5 Lexicon Model

Similar to (Vogel et al., 2004), we compute for each candidate block a score within a given sentence pair using a word level lexicon $P(f|e)$ as follows:

$$\begin{aligned} P(f_j^{j+l}|e_i^{i+k}) &= \prod_{j' \in [j,j+l]} \sum_{i' \in [i,i+k]} \frac{P(f_{j'}|e_{i'})}{k+1} \\ &\cdot \prod_{j' \notin [j,j+l]} \sum_{i' \notin [i,i+k]} \frac{P(f_{j'}|e_{i'})}{|E|-k-1} \end{aligned}$$

6 Algorithm

Our phrase extraction is described in Algorithm 1. The input parameters are essentially from IBM Model-4: the word level lexicon $P(f|e)$, the English word level fertility $P_\phi(\phi_e = k|e)$, and the center based distortion $P(\odot_{e_i^{i+k}}|\odot_{f_j^{j+l}})$.

Overall, for each source phrase f_j^{j+l}, the algorithm first estimates its normalized relative center in the source sentence, its projected relative center in the target sentence. The scores of the phrase length, center-based distortion, and a lexicon based score are computed for each candidate block A local greedy search is carried out for the best scored phrase pair (f_j^{j+l}, e_i^{i+k}).

In our submitted system, we computed the following *seven* base scores for phrase pairs: $P_{ef}(f_j^{j+l}|e_i^{i+k})$, $P_{fe}(e_i^{i+k}|f_j^{j+l})$, sharing similar function form in Eqn. 5.

$$\begin{aligned} P_{ef}(f_j^{j+l}|e_i^{i+k}) &= \prod_{j'} \sum_{i'} P(f_{j'}|e_{i'}) P(e_{i'}|e_i^{i+k}) \\ &= \prod_{j'} \sum_{i'} \frac{P(f_{j'}|e_{i'})}{k+1} \end{aligned} \tag{5}$$

We compute phrase level relative frequency in both directions: $P_{rf}(f_j^{j+l}|e_i^{i+k})$ and $P_{rf}(e_i^{i+k}|f_j^{j+l})$. We compute two other lexicon scores which were also used in (Vogel et al., 2004): $S_1(f_j^{j+l}|e_i^{i+k})$ and $S_2(e_i^{i+k}|f_j^{j+l})$ using the similar function in Eqn. 6:

$$S(f_j^{j+l}|e_i^{i+k}) = \prod_{j'} \sum_{i'} P(f_{j'}|e_{i'}) \tag{6}$$

In addition, we put the *phrase level fertility score* computed in section 3 via dynamic programming to be as one additional score for decoding.

Algorithm 1 A Generalized Alignment-free Phrase Extraction

1: **Input**: Pre-trained models: $P_\phi(\phi_e = k|e)$, $P(\odot_E|\odot_F)$, and $P(f|e)$.
2: **Output**: PhraseSet: Phrase pair collections.
3: **Loop** over the next sentence pair
4: for $j : 0 \rightarrow |F| - 1$,
5: for $l : 0 \rightarrow$ MaxLength,
6: foreach f_j^{j+l}
7: compute $\odot_f$ and $\odot_E$
8: left = $\odot_E \cdot |E|$-MaxLength,
9: right= $\odot_E \cdot |E|$+MaxLength,
10: for $i :$ left $\rightarrow$ right,
11: for $k : 0 \rightarrow$ right,
12: compute $\odot_e$ of e_i^{i+k},
13: score the phrase pair (f_j^{j+l}, e_i^{i+k}), where score = $P(\odot_e|\odot_f)P(l|e_i^{i+k})P(f_j^{j+l}|e_i^{i+k})$
14: add top-n $\{(f_j^{j+l}, e_i^{i+k})\}$ into PhraseSet.

7 Experimental Results

Our system is based on the IBM Model-4 parameters. We train IBM Model 4 with a scheme of $1^7 2^0 h^7 3^0 4^3$ using GIZA++ (Och and Ney, 2003). The maximum fertility for an English word is 3. All the data is used as given, i.e. we do not have any preprocessing of the English-French data. The word alignment provided in the workshop is not used in our evaluations. The language model is provided by the workshop, and we do not use other language models.

The French phrases up to 8-gram in the development and test sets are extracted with top-3 candidate English phrases. There are in total 2.6 million phrase pairs [1] extracted for both development set and the unseen test set. We did minimal tuning of the parameters in the pharaoh decoder (Koehn, 2004) settings, simply to balance the length penalty for Bleu score. Most of the weights are left as they are given: [ttable-limit]=20, [ttable-threshold]=0.01, [stack]=100, [beam-threshold]=0.01, [distortion-limit]=4, [weight-d]=0.5, [weight-l]=1.0, [weight-w]=-0.5. Table 1 shows the algorithm's performance on several settings for the *seven* basic scores provided in section 6.

settings	Dev.Bleu	Tst.Bleu
s_1	27.44	27.65
s_2	27.62	28.25

Table 1: Pharaoh Decoder Settings

In Table 1, setting s_1 was our submission without using the inverse relative frequency of $P_{rf}(e_i^{i+k}|f_j^{j+l})$. s_2 is using all the seven scores.

8 Discussions

In this paper, we propose a generalized phrase extraction algorithm towards word alignment-free utilizing the fertility model to predict the width of the block, a distortion model to predict how close the centers of source and target phrases are, and a lexicon model for translational equivalence. The algorithm is a general framework, in which one could plug in other scores and word alignment to get better results.

References

P.F. Brown, Stephen A. Della Pietra, Vincent. J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

Philip Koehn. 2004. Pharaoh: a beam search decoder for phrase-based smt. In *Proceedings of the Conference of the Association for Machine Translation in the Americans (AMTA)*.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29, pages 19–51.

Christoph Tillmann. 2003. A projection extension algorithm for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Stephan Vogel, Sanjika Hewavitharana, Muntsin Kolss, and Alex Waibel. 2004. The ISL statistical translation system for spoken language translation. In *Proc. of the International Workshop on Spoken Language Translation*, pages 65–72, Kyoto, Japan.

[1] Our phrase table is to be released to public in this workshop

Combining Linguistic Data Views for Phrase-based SMT

Jesús Giménez and **Lluís Màrquez**
TALP Research Center, LSI Department
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1–3, E-08034, Barcelona
{jgimenez,lluism}@lsi.upc.edu

Abstract

We describe the Spanish-to-English *LDV-COMBO* system for the Shared Task 2: "Exploiting Parallel Texts for Statistical Machine Translation" of the ACL-2005 Workshop on "Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond". Our approach explores the possibility of working with alignments at different levels of abstraction, using different degrees of linguistic annotation. Several phrase-based translation models are built out from these alignments. Their combination significatively outperforms any of them in isolation. Moreover, we have built a word-based translation model based on WordNet which is used for unknown words.

1 Introduction

The main motivation behind our work is to introduce linguistic information, other than lexical units, to the process of building word and phrase alignments. Many other authors have tried to do so. See (Och and Ney, 2000), (Yamada and Knight, 2001), (Koehn and Knight, 2002), (Koehn et al., 2003), (Schafer and Yarowsky, 2003) and (Gildea, 2003).

Far from full syntactic complexity, we suggest to go back to the simpler alignment methods first described by (Brown et al., 1993). Our approach exploits the possibility of working with alignments at two different levels of granularity, lexical (words) and shallow parsing (chunks). In order to avoid confusion so forth we will talk about *tokens* instead of *words* as the minimal alignment unit.

Apart from redefining the scope of the alignment unit, we may use different degrees of linguistic annotation. We introduce the general concept of *data view*, which is defined as any possible representation of the information contained in a bitext. We enrich data view tokens with features further than lexical such as *PoS*, *lemma*, and *chunk label*.

As an example of the applicability of data views, suppose the case of the word *'plays'* being seen in the training data acting as a verb. Representing this information as *'plays*$_{VBZ}$*'* would allow us to distinguish it from its homograph *'plays*$_{NNS}$*'* for *'plays'* as a noun. Ideally, one would wish to have still deeper information, moving through syntax onto semantics, such as *word senses*. Therefore, it would be possible to distinguish for instance between two realizations of *'plays'* with different meanings: *'he*$_{PRP}$ *plays*$_{VBG}$ *guitar*$_{NN}$*'* and *'he*$_{PRP}$ *plays*$_{VBG}$ *basketball*$_{NN}$*'*.

Of course, there is a natural trade-off between the use of data views and data sparsity. Fortunately, we hava data enough so that statistical parameter estimation remains reliable.

2 System Description

The *LDV-COMBO* system follows the SMT architecture suggested by the workshop organizers.

First, training data are linguistically annotated for the two languages involved (See subsection 2.1). 10 different data views have been built. Notice that it is not necessary that the two parallel counterparts of a bitext share the same data view, as

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 145–148,
Ann Arbor, June 2005.

long as they share the same granularity. However, in all our experiments we have annotated both sides with the same linguistic information. See token descriptions: (W) word, (WL) word and lemma, (WP) word and PoS, (WC) word and chunk label, (WPC) word, PoS and chunk label, (Cw) chunk of words (Cwl), chunk of words and lemmas, (Cwp) chunk of words and PoS (Cwc) chunk of words and chunk labels (Cwpc) chunk of words, PoS and chunk labels. By chunk label we refer to the IOB label associated to every word inside a chunk, e.g. *'I_{B-NP} $declare_{B-VP}$ $resumed_{I-VP}$ the_{B-NP} $session_{I-NP}$ of_{B-PP} the_{B-NP} $European_{I-NP}$ $Parliament_{I-NP}$ $._{O}$'*). We build chunk tokens by explicitly connecting words in the same chunk, e.g. *'$(I)_{NP}$ $(declare\_resumed)_{VP}$ $(the\_session)_{NP}$ $(of)_{PP}$ $(the\_European\_Parliament)_{NP}$'*. See examples of some of these data views in Table 1.

Then, running *GIZA++*, we obtain token alignments for each of the data views. Combined phrase-based translation models are built on top of the Viterbi alignments output by *GIZA++*. See details in subsection 2.2. *Combo-models* must be then post-processed in order to remove the additional linguistic annotation and split chunks back into words, so they fit the format required by *Pharaoh*.

Moreover, we have used the Multilingual Central Repository (MCR), a multilingual lexical-semantic database (Atserias et al., 2004), to build a word-based translation model. We back-off to this model in the case of unknown words, with the goal of improving system recall. See subsection 2.3.

2.1 Data Representation

In order to achieve robustness the same tools have been used to linguistically annotate both languages. The *SVMTool*[1] has been used for PoS-tagging (Giménez and Màrquez, 2004). The *Freeling*[2] package (Carreras et al., 2004) has been used for lemmatizing. Finally, the *Phreco* software by (Carreras et al., 2005) has been used for shallow parsing.

No additional tokenization or pre-processing steps other than case lowering have been performed. Special treatment of named entities, dates, numbers, currency, etc., should be considered so as to further enhance the system.

2.2 Building Combined Translation Models

Because data views capture different, possibly complementary, aspects of the translation process it seems reasonable to combine them. We consider two different ways of building such combo-models:

LPHEX Local phrase extraction. To build a separate phrase-based translation model for each data view alignment, and then combine them. There are two ways of combining translation models:

- **MRG** Merging translation models. We work on a weighted linear interpolation of models. These weights may be tuned, although a uniform weight selection yields good results. Additionally, phrase-pairs may be filtered out by setting a score threshold.
- **noMRG** Passing translation models directly to the Pharaoh decoder. However, we encountered many problems with phrase-pairs that were not seen in all single models. This obliged us to apply arbitrary smoothing values to score these pairs.

GPHEX Global phrase extraction. To build a single phrased-based translation model from the union of alignments from several data views.

In its turn, any MRG operation performed on a combo-model results again in a valid combo-model.

In any case, phrase extraction[3] is performed as depicted by (Och, 2002).

2.3 Using the MCR

Outer knowledge may be supplied to the *Pharaoh* decoder by annotating the input with alternative translation options via XML-markup. We enrich every unknown word by looking up every possible translation for all of its senses in the MCR. These are scored by relative frequency according to the number of senses that lexicalized in the same manner. Let w_f, p_f be the source word and PoS, and w_e be the target word, we define a function

[1] The SVMTool may be freely downloaded at `http://www.lsi.upc.es/~nlp/SVMTool/`.

[2] Freeling Suite of Language Analyzers may be downloaded at `http://www.lsi.upc.es/~nlp/freeling/`

[3] We always work with the union of alignments, no heuristic refinement, and phrases up to 5 tokens. Phrase pairs appearing only once have been discarded. Scoring is performed by relative frequency. No smoothing is applied.

WPC	It$_{[PRP:B-NP]}$ would$_{[MD:B-VP]}$ appear$_{[VB:I-VP]}$ that$_{[IN:B-SBAR]}$ a$_{[DT:B-NP]}$ speech$_{[NN:I-NP]}$ made$_{[VBN:B-VP]}$ at$_{[IN:B-PP]}$ the$_{[DT:B-NP]}$ weekend$_{[NN:I-NP]}$ by$_{[IN:B-PP]}$ Mr$_{[NNP:B-NP]}$ Fischler$_{[NNP:I-NP]}$ indicates$_{[VBZ:B-VP]}$ a$_{[DT:B-NP]}$ change$_{[NN:I-NP]}$ of$_{[IN:B-PP]}$ his$_{[PRP\$:B-NP]}$ position$_{[NN:I-NP]}$.$_{[.:O]}$ Fischler$_{[VMN:B-VP]}$ pronunció$_{[VMI:B-VP]}$ un$_{[DI:B-NP]}$ discurso$_{[NC:I-NP]}$ este$_{[DD:B-NP]}$ fin$_{[NC:I-NP]}$ de$_{[SP:B-PP]}$ semana$_{[NC:B-NP]}$ en$_{[SP:B-PP]}$ el$_{[DA:B-SBAR]}$ que$_{[PR0:I-SBAR]}$ parecía$_{[VMI:B-VP]}$ haber$_{[VAN:I-VP]}$ cambiado$_{[VMP:I-VP]}$ de$_{[SP:B-PP]}$ actitud$_{[NC:B-NP]}$.$_{[Fp:O]}$
Cwpc	(It$_{[PRP:B-NP]}$) (would$_{[MD:B-VP]}$_appear$_{[VB:I-VP]}$) (that$_{[IN:B-SBAR]}$) (a$_{[DT:B-NP]}$_speech$_{[NN:I-NP]}$) (made$_{[VBN:B-VP]}$) (at$_{[IN:B-PP]}$) (the$_{[DT:B-NP]}$_weekend$_{[NN:I-NP]}$) (by$_{[IN:B-PP]}$) (Mr$_{[NNP:B-NP]}$_Fischler$_{[NNP:I-NP]}$) (indicates$_{[VBZ:B-VP]}$) (a$_{[DT:B-NP]}$_change$_{[NN:I-NP]}$) (of$_{[IN:B-PP]}$) (his$_{[PRP\$:B-NP]}$_position$_{[NN:I-NP]}$) (.$_{[.:O]}$) (Fischler$_{[VMN:B-VP]}$) (pronunció$_{[VMI:B-VP]}$) (un$_{[DI:B-NP]}$_discurso$_{[NC:I-NP]}$) (este$_{[DD:B-NP]}$_fin$_{[NC:I-NP]}$) (de$_{[SP:B-PP]}$) (semana$_{[NC:B-NP]}$) (en$_{[SP:B-PP]}$) (el$_{[DA:B-SBAR]}$_que$_{[PR0:I-SBAR]}$) (parecía$_{[VMI:B-VP]}$_haber$_{[VAN:I-VP]}$_cambiado$_{[VMP:I-VP]}$) (de$_{[SP:B-PP]}$) (actitud$_{[NC:B-NP]}$) (.$_{[Fp:O]}$)

Table 1: An example of 2 rich data views: (WPC) word, PoS and IOB chunk label (Cwpc) chunk of word, PoS and chunk label.

$Scount(w_f, p_f, w_e)$ which counts the number of senses for (w_f, p_f) which can lexicalize as w_e. A translation pair is scored as:

$$score(w_f, p_f|w_e) = \frac{Scount(w_f, p_f, w_e)}{\sum_{(w_f,p_f)} Scount(w_f, p_f, w_e)} \quad (1)$$

Better results would be expected working with word sense disambiguated text. We are not at this point yet. A first approach could be to work with the most frequent sense heuristic.

3 Experimental Results

3.1 Data and Evaluation Metrics

We have used the data sets and language model provided by the organization. No extra training or development data were used in our experiments.

We evaluate results with 3 different metrics: GTM F_1-measure ($e = 1, 2$), BLEU score ($n = 4$) as provided by organizers, and NIST score ($n = 5$).

3.2 Experimenting with Data Views

Table 2 presents MT results for the 10 elementary data views devised in Section 2. Default parameters are used for λ_{tm}, λ_{lm}, and λ_w. No tuning has been performed. As expected, word-based views obtain significatively higher results than chunk-based. All data views at the same level of granularity obtain comparable results.

In Table 3 MT results for different data view combinations are showed. Merged model weights are set equiprobable, and no phrase-pair score filtering is performed. We refer to the W model as our baseline. In this view, only words are used. The 5W-MRG and 5W-GPHEX models use a combination of the 5 word-based data views, as in MRG and GPHEX, respectively. The 5C-MRG and 5C-GPHEX system use a combination of the 5 chunk based data views, as in MRG and GPHEX, respectively. The 10-MRG system uses all 10 data views combined as in MRG. The 10-GPHEX/MRG system uses the 5 word based views combined as in GPHEX, the 5 chunk based views combined as in GPHEX, and then a combination of these two combo-models as in MRG.

data view	GTM-1	GTM-2	BLEU	NIST
W	0.6108	0.2609	25.92	7.1576
WL	0.6110	0.2601	25.77	7.1496
WP	0.6096	0.2600	25.74	7.1415
WC	0.6124	0.2600	25.98	7.1852
WPC	0.6107	0.2587	25.79	7.1595
Cw	0.5749	0.2384	22.73	6.6149
Cwl	0.5756	0.2385	22.73	6.6204
Cwp	0.5771	0.2395	23.06	6.6403
Cwc	0.5759	0.2390	22.86	6.6207
Cwpc	0.5744	0.2379	22.77	6.5949

Table 2: MT Results for the 10 elementary data views on the development set.

data view	GTM-1	GTM-2	BLEU	NIST
W	0.6108	0.2609	25.92	7.1576
5W-MRG	0.6134	0.2631	26.25	7.2122
5W-GPHEX	0.6172	0.2615	26.95	7.2823
5C-MRG	0.5786	0.2407	23.18	6.6754
5C-GPHEX	0.5739	0.2368	22.80	6.5714
10-MRG	0.6130	0.2624	26.24	7.2196
10-GPHEX/MRG	0.6142	0.2600	26.58	7.2542

Table 3: MT Results without tuning, for some data view combinations on the development set.

It can be seen that results improve by combining several data views. Furthermore, global phrase extraction (GPHEX) seems to work much finer than local phrase extraction (LPHEX).

Table 4 shows MT results after optimizing λ_{tm}, λ_{lm}, λ_w, and the weights for the MRG operation, by means of the *Downhill Simplex Method in Multidimensions* (William H. Press and Flannery, 2002). Observe that tuning the system improves the performance considerably. The λ_w parameter is particularly sensitive to tuning.

Even though the performance of chunk-based models is poor, the best results are obtained by combinining the two levels of abstraction, thus proving that syntactically motivated phrases may help. 10-MRG and 10-GPHEX models achieve a similar performance. The *10-MRG-best*$_{WN}$ system corresponds to the 10-MRG model using WordNet. The *10-MRG-sub*$_{WN}$ system is this same system at the time of submission. Results using WordNet, taking into account that the number of unknown[4] words in the development set was very small, are very promising.

data view	GTM-1	GTM-2	BLEU	NIST
W	0.6174	0.2583	28.13	7.1540
5W-MRG	0.6206	0.2605	28.50	7.2076
5W-GPHEX	0.6207	0.2603	28.38	7.1992
5C-MRG	0.5882	0.2426	25.06	6.6773
5C-GPHEX	0.5816	0.2387	24.40	6.5595
10-MRG	0.6218	0.2623	28.88	7.2491
10-GPHEX/MRG	0.6229	0.2622	28.82	7.2414
10-MRG$_{WN}$	0.6228	0.2625	28.90	7.2583
10-MRG-sub$_{WN}$	0.6228	0.2622	28.79	7.2528

Table 4: MT Results for some data view combinations after tuning on the development set.

4 Conclusions

We have showed that it is possible to obtain better phrase-based translation models by utilizing alignments built on top of different linguistic data views. These models can be robustly combined, significantly outperforming all of their components in isolation. We leave for further work the experimentation of new data views such as word senses and semantic roles, as well as their natural porting and evolution from the alignment step to phrase extraction and decoding.

[4]Translation for 349 unknown words was found in the MCR.

Acknowledgements

This research has been funded by the Spanish Ministry of Science and Technology (ALIADO TIC2002-04447-C02). Authors are thankful to Patrik Lambert for providing us with the implementation of the Simplex Method used for tuning.

References

Jordi Atserias, Luis Villarejo, German Rigau, Eneko Agirre, John Carroll, Bernardo Magnini, and Piek Vossen. 2004. The meaning multilingual central repository. In *Proceedings of GWC*, Brno, Czech Republic, January. ISBN 80-210-3302-9.

Peter E Brown, Stephen A. Della Pietra, Robert L. Mercer, and Vincent J. Della Pietra. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th LREC*.

Xavier Carreras, Lluís Márquez, and Jorge Castro. 2005. Filtering-ranking perceptron learning for partial parsing. *Machine Learning*, 59:1–31.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of ACL*.

Jesús Giménez and Lluís Màrquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of 4th LREC*.

Philipp Koehn and Kevin Knight. 2002. Chunkmt: Statistical machine translation with richer linguistic knowledge. Draft.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*.

Franz Josef Och. 2002. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph.D. thesis, RWTH Aachen, Germany.

Charles Schafer and David Yarowsky. 2003. Statistical machine translation using coercive two-level syntactic transduction. In *Proceedings of EMNLP*.

William T. Vetterling William H. Press, Saul A. Teukolsky and Brian P. Flannery. 2002. *Numerical Recipes in C++: the Art of Scientific Computing*. Cambridge University Press.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL*.

Improving Phrase-Based Statistical Translation by modifying phrase extraction and including several features

Marta Ruiz Costa-jussà and José A. R. Fonollosa

TALP Research Center
Universitat Politècnica de Catalunya
{mruiz,adrian}@gps.tsc.upc.edu

Abstract

Nowadays, most of the statistical translation systems are based on phrases (i.e. groups of words). In this paper we study different improvements to the standard phrase-based translation system. We describe a modified method for the phrase extraction which deals with larger phrases while keeping a reasonable number of phrases. We also propose additional features which lead to a clear improvement in the performance of the translation. We present results with the EuroParl task in the direction Spanish to English and results from the evaluation of the shared task "Exploiting Parallel Texts for Statistical Machine Translation" (ACL Workshop on Parallel Texts 2005).

1 Introduction

Statistical Machine Translation (SMT) is based on the assumption that every sentence e in the target language is a possible translation of a given sentence f in the source language. The main difference between two possible translations of a given sentence is a probability assigned to each, which has to be learned from a bilingual text corpus. Thus, the translation of a source sentence f can be formulated as the search of the target sentence e that maximizes the translation probability $P(e|f)$,

$$\tilde{e} = \underset{e}{argmax}\, P(e|f) \tag{1}$$

[0]This work has been supported by the European Union under grant FP6-506738 (TC-STAR project).

If we use Bayes rule to reformulate the translation probability, we obtain,

$$\tilde{e} = \underset{e}{argmax}\, P(f|e)P(e) \tag{2}$$

This translation model is known as the source-channel approach [1] and it consists on a language model $P(e)$ and a separate translation model $P(f|e)$ [5].

In the last few years, new systems tend to use sequences of words, commonly called phrases [8], aiming at introducing word context in the translation model. As alternative to the source-channel approach the decision rule can be modeled through a log-linear maximum entropy framework.

$$\tilde{e} = \underset{e}{argmax} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e, f) \right\} \tag{3}$$

The features functions, h_m, are the system models (translation model, language model and others) and weigths, λ_i, are typically optimized to maximize a scoring function. It is derived from the Maximum Entropy approach suggested by [13] [14] for a natural language understanding task. It has the advantatge that additional features functions can be easily integrated in the overall system.

This paper addresses a modification of the phrase-extraction algorythm in [11]. It also combines several interesting features and it reports an important improvement from the baseline. It is organized as follows. Section 2 introduces the baseline; the following section explains the modification in the phrase extraction; section 4 shows the different features which have been taken into account; section 5 presents the evaluation framework; and

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 149–154,
Ann Arbor, June 2005.

the final section shows some conclusions on the experiments in the paper and on the results in the shared task.

2 Baseline

The baseline is based on the source-channel approach, and it is composed of the following models which later will be combined in the decoder.

The Translation Model. It is based on bilingual phrases, where a bilingual phrase (BP) is simply two monolingual phrases (MP) in which each one is supposed to be the translation of each other. A monolingual phrase is a sequence of words. Therefore, the basic idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations [17].

During training, the system has to learn a dictionary of phrases. We begin by aligning the training corpus using GIZA++ [6], which is done in both translation directions. We take the union of both alignments to obtain a symmetrized word alignment matrix. This alignment matrix is the starting point for the phrase based extraction.

Next, we define the criterion to extract the set of BP of the sentence pair $(f_{j_1}^{j_2}; e_{i_1}^{i_2})$ and the alignment matrix $A \subseteq J*I$, which is identical to the alignment criterion described in [11].

$$BP(f_1^J, e_1^I, A) = \{(f_{j_1}^{j_2}, e_{i_1}^{i_2}) :$$

$$\forall (j,i) \epsilon A : j_1 \leq j \leq j_2 \leftrightarrow i_1 \leq i \leq i2$$

$$\wedge \exists (j,i) \epsilon A : j_1 \leq j \leq j_2 \wedge i_1 \leq i \leq i2\}$$

The set of BP is consistent with the alignment and consists of all BP pairs where all words within the foreign language phrase are only aligned to the words of the English language phrase and viceversa. At least one word in the foreign language phrase has to be aligned with at least one word of the English language. Finally, the algorithm takes into account possibly unaligned words at the boundaries of the foreign or English language phrases.

The target language model. It is combined with the translation probability as showed in equation (2). It gives coherence to the target text obtained by the concatenated phrases.

3 Phrase Extraction

Motivation. The length of a MP is defined as its number of words. The length of a BP is the greatest of the lengths of its MP.

As we are working with a huge amount of data (see corpus statistics), it is unfeasible to build a dictionary with all the phrases longer than length 4. Moreover, the huge increase in computational and storage cost of including longer phrases does not provide a significant improve in quality [8].

X-length In our system we considered two length limits. We first extract all the phrases of length 3 or less. Then, we also add phrases up to length 5 if they cannot be generated by smaller phrases. Empirically, we chose 5, as the probability of reappearence of larger phrases decreases.

Basically, we select additional phrases with source words that otherwise would be missed because of cross or long alignments. For example, from the following sentence,

Cuando el Parlamento Europeo , que tan frecuentemente insiste en los derechos de los trabajadores y en la debida protección social , (...)

NULL ( ) When (1) the (2) European (4) Parliament (3 4) , (5) that (6) so (7) frequently (8) insists (9) on (10) workers (11 15) ' (14) rights (12) and (16) proper (19) social (21) protection (20) , (22) (...)

where the number inside the clauses is the aligned word(s). And the phrase that we are looking for is the following one.

los derechos de los trabajadores # workers ' rights

which only could appear in the case the maximum length was 5.

4 Phrase ranking

4.1 Conditional probability $P(f|e)$

Given the collected phrase pairs, we estimated the phrase translation probability distribution by relative frecuency.

$$P(f|e) = \frac{N(f,e)}{N(e)} \quad (4)$$

where N(f,e) means the number of times the phrase f is translated by e. If a phrase e has $N > 1$ possible translations, then each one contributes as *1/N* [17].

Note that no smoothing is performed, which may cause an overestimation of the probability of rare phrases. This is specially harmful given a BP where the source part has a big frecuency of appearence but the target part appears rarely. For example, from our database we can extract the following BP: "you # la que no", where the English is the source language and the Spanish, the target language. Clearly, "la que no" is not a good translation of "you", so this phrase should have a low probability. However, from our aligned training database we obtain,

$$P(f|e) = P(\text{you}|\text{la que no}) = 0.23$$

This BP is clearly overestimated due to sparseness. On the other, note that "la que no" cannot be considered an unusual trigram in Spanish. Hence, the language model does not penalise this target sequence either. So, the total probability $(P(f|e)P(e))$ would be higher than desired.

In order to somehow compensate these unreiliable probabilities we have studied the inclusion of the posterior [12] and lexical probabilities [1] [10] as additional features.

4.2 Feature $P(e|f)$

In order to estimate the posterior phrase probability, we compute again the relative frequency but replacing the count of the target phrase by the count of the source phrase.

$$P(e|f) = \frac{N'(f,e)}{N(f)} \quad (5)$$

where N'(f,e) means the number of times the phrase e is translated by f. If a phrase f has $N > 1$ possible translations, then each one contributes as *1/N*.

Adding this feature function we reduce the number of cases in which the overall probability is overestimated. This results in an important improvement in translation quality.

4.3 IBM Model 1

We used IBM Model 1 to estimate the probability of a BP. As IBM Model 1 is a word translation and it gives the sum of all possible alignment probabilities, a lexical co-ocurrence effect is expected. This captures a sort of semantic coherence in translations.

Therefore, the probability of a sentence pair is given by the following equation.

$$P(f|e; M1) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(f_j|e_i) \quad (6)$$

The $p(f_j|e_i)$ are the source-target IBM Model 1 word probabilities trained by GIZA++. Because the phrases are formed from the union of source-to-target and target-to-source alignments, there can be words that are not in the $P(f_j|e_i)$ table. In this case, the probability was taken to be 10^{-40}.

In addition, we have calculated the IBM^{-1} Model 1.

$$P(e|f; M1) = \frac{1}{(J+1)^I} \prod_{I=1}^{I} \sum_{j=0}^{J} p(e_i|f_j) \quad (7)$$

4.4 Language Model

The English language model plays an important role in the source channel model, see equation (2), and also in its modification, see equation (3). The English language model should give an idea of the sentence quality that is generated.

As default language model feature, we use a standard word-based trigram language model generated with smoothing Kneser-Ney and interpolation (by using SRILM [16]).

4.5 Word and Phrase Penalty

To compensate the preference of the target language model for shorter sentences, we added two

	Spanish	*English*
Train Sentences	1223443	1223443
Words	34794006	33379333
Vocabulary	168685	104975
Dev Sentences	504	504
Words	15353	15335
OOV	25	16
Test Sentences	504	504
Words	10305	10667
OOV	36	19

Table 1: *Statistics of training and test corpus*

simple features which are widely used [17] [7]. The word penalty provides means to ensure that the translations do not get too long or too short. Negative values for the word penalty favor longer output, positive values favor shorter output [7].

The phrase penalty is a constant cost per produced phrase. Here, a negative weight, which means reducing the costs per phrase, results in a preference for adding phrases. Alternatively, by using a positive scaling factors, the system will favor less phrases.

5 Evaluation framework

5.1 Corpus Statistics

Experiments were performed to study the effect of our modifications in the phrases. The training material covers the transcriptions from April 1996 to September 2004. This material has been distributed by the European Parlament. In our experiments, we have used the distribution of RWTH of Aachen under the project of TC-STAR [1]. The test material was used in the first evaluation of the project in March 2005. In our case, we have used the development divided in two sets. This material corresponds to the transcriptions of the sessions from October the 21st to October the 28th. It has been distributed by ELDA[2]. Results are reported for Spanish-to-English translations.

[1] http://www.tcstar.org/
[2] http://www.elda.org/

5.2 Experiments

The decoder used for the presented translation system is reported in [2]. This decoder is called MARIE and it takes into account simultaneously all the 7 features functions described above. It implements a beam-search strategy.

As evaluation criteria we use: the Word Error Rate (WER), the BLEU score [15] and the NIST score [3].

As follows we report the results for several experiments that show the performance of: the baseline, adding the posterior probability, IBM Model 1 and IBM1^{-1}, and, finally, the modification of the phrases extraction.

Optimisation. Significant improvements can be obtained by tuning the parameters of the features adequately. In the complet system we have 7 parameters to tune: the relatives frecuencies $P(f|e)$ and $P(e|f)$, IBM Model 1 and its inverse, the word penalty, the phrase penalty and the weight of the language model. We applied the widely used algorithm SIMPLEX to optimise [9]. In Table 2 (line 5th), we see the final results.

Baseline. We report the results of the baseline. We use the union alignment and we extract the BP of length 3. As default language model feature, we use the standard trigram with smoothing Kneser-Ney and interpolation. Also we tune the parameters (only two parameters) with the SIMPLEX algorithm (see Table 2).

Posterior probability. Table 2 shows the effect of using the posterior probability: $P(e|f)$. We use all the features but the $P(e|f)$ and we optimise the parameters. We see the results without this feature decrease around 1.1 points both in BLEU and WER (see line 2rd and 5th in Table 2).

IBM Model 1. We do the same as in the paragraph above, we do not consider the IBM Model 1 and the IBM1^{-1}. Under these conditions, the translation's quality decreases around 1.3 points both in BLEU and WER (see line 3th and 5th in Table 2).

Modification of the Phrase Extraction. Finally, we made an experiment without modification of the phrases' length. We can see the comparison between: (1) the phrases of fixed maximum length of 3; and (2) including phrases with a maximum length of 5 which can not be generated by smaller phrases. We can see it in Table 2 (lines 4th and 5th). We observe that there is no much difference between the number of phrases, so this approach does not require more resources. However, we get slightly better scores.

5.3 Shared Task

This section explains the participation of "Exploiting Parallel Texts for Statistical Machine Translation". We used the EuroParl data provided for this shared task [4]. A word-to-word alignment was performed in both directions as explained in section 2. The phrase-based translation system which has been considered implements a total of 7 features (already explained in section 4). Notice that the language model has been trained with the training provided in the shared task. However, the optimization in the parameters has not been repeated, and we used the parameters obtained in the subsection above. We have obtained the results in the Table 3.

6 Conclusions

We reported a new method to extract longer phrases without increasing the quantity of phrases (less than 0.5%).

We also reported several features as $P(e|f)$ which in combination with the functions of the source-channel model provides significant improvement. Also, the feature IBM1 in combination with IBM1^{-1} provides improved scores, too.

Finally, we have optimized the parameters, and we provided the final results which have been presented in the Shared Task: Exploiting Parallel Texts for Statistical Machine Translation (June 30, 2005) in conjunction with ACL 2005 in Ann Arbor, Michigan.

7 Acknowledgements

The authors want to thank José B. Mariño, Adrià de Gispert, Josep M. Crego, Patrik Lambert and Rafael E. Banchs (members of the TALP Research Center) for their contribution to this work.

References

[1] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June 1990.

[2] Josep M. Crego, José B. Mariño, and Adrià de Gispert. An Ngram-based Statistical Machine Translation Decoder. In *Draft*, 2005.

[3] G. Doddington. Automatic evaluation machine translation quality using n-gram co-ocurrence statistics. In *Proc. ARPA Workshop on Human Language Technology*, 2002.

[4] EuroParl: European Parliament Proceedings Parallel Corpus. Available on-line at: http://people.csail.mit.edu/koehn/publications/europarl/. 1996-2003.

[5] I. García-Varea. *Traducción Automática estadística: Modelos de Traducción basados en Máxima Entropía y Algoritmos de Búsqueda* . UPV, Diciembre 2003.

[6] Giza++. http://www-i6.informatik.rwth-aachen.de/∼och/software/giza++.html/, 1999.

[7] P. Koehn. A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. 2003.

[8] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, pages 127–133, May 2003.

[9] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

Phr Length	λ_{LM}	$\lambda_{p(f\|e)}$	$\lambda_{p(e\|f)}$	λ_{IBM1}	$\lambda_{IBM1^{-1}}$	λ_{PP}	λ_{WP}	*WER*	*BLEU*	*NIST*	*# frases*
3	0.788	0.906	0	0	0	0	0	33.98	57.44	10.11	67.7M
3+5length	0.788	0.941	0	0.771	0.200	3.227	0.448	28.97	64.71	11.07	68M
3+5length	0.788	0.824	0.820	0	0	3.430	-0.083	29.17	64.59	10.99	68M
3	0.746	0.515	0.979	0.514	0.390	1.537	-1.264	27.94	65.70	11.18	67.7M
3+5length	0.788	0.617	0.810	0.635	0.101	1.995	-0.296	27.88	65.82	11.23	68M

Table 2: *Results for the different experiments with optimized parameters in the direction SPA->ENG*

Phr Length	λ_{LM}	$\lambda_{p(f\|e)}$	$\lambda_{p(e\|f)}$	λ_{IBM1}	$\lambda_{IBM1^{-1}}$	λ_{PP}	λ_{WP}	*BLEU*	*# frases*
3+5length	0.788	0.617	0.810	0.635	0.101	1.995	-0.296	29.84	34.8M

Table 3: *Results for the ACL training and ACL test (SPA->ENG)*

[10] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, May 2004.

[11] F. J. Och and H. Ney. The Alignment Template Approach to Statistical Machine Translation. *Computational linguistics*, 30:417–449, December 2004.

[12] Franz Josef Och and Hermann Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *ACL*, pages pages 295–302, July 2002.

[13] Papineni, S.Roukos, and R.T. Ward. Feature-based language understanding. In *European Conf. on Speech Communication and Technology*, pages 1435–1438, September 1997.

[14] Papineni, S.Roukos, and R.T. Ward. Maximum likelihood and discriminative training of direct translation models. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Proceedings*, pages 189–192, May 1998.

[15] K.A. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022), IBM Research Division*, 2001.

[16] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings Intl. Conference Spoken Language Processing*, September 2002.

[17] R. Zens and H. Ney. Improvements in Phrase-Based Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, pages 257–264, May 2004.

First Steps towards Multi-Engine Machine Translation

Andreas Eisele
Computational Linguistics
Saarland University P.O.Box 151150
D-66041 Saarbrücken, Germany
eisele@coli.uni-saarland.de

Abstract

We motivate our contribution to the shared MT task as a first step towards an integrated architecture that combines advantages of statistical and knowledge-based approaches. Translations were generated using the Pharaoh decoder with tables derived from the provided alignments for all four languages, and for three of them using web-based and locally installed commercial systems. We then applied statistical and heuristic algorithms to select the most promising translation out of each set of candidates obtained from a source sentence. Results and possible refinements are discussed.

1 Motivation and Long-term Perspective

"The problem of robust, efficient and reliable speech-to-speech translation can only be cracked by the combined muscle of deep and shallow processing approaches." (Wahlster, 2001) Although this statement has been coined in the context of VerbMobil, aiming at translation for direct communication, it appears also realistic for many other translation scenarios, where demands on robustness, coverage, or adaptability on the input side and quality on the output side go beyond today's technological possibilities. The increasing availability of MT engines and the need for better quality has motivated considerable efforts to combine multiple engines into one "super-engine" that is hopefully better than any of its ingredients, an idea pionieered in (Frederking and Nirenburg, 1994). So far, the larger group of related publications has focused on the task of selecting, from a set of translation candidates obtained from different engines, one translation that looks most promising (Tidhar and Küssner, 2000; Akiba et al., 2001; Callison-Burch and Flournoy, 2001; Akiba et al., 2002; Nomoto, 2004). But also the more challenging problem of decomposing the candidates and re-assembling from the pieces a new sentence, hopefully better than any of the given inputs, has recently gained considerable attention (Rayner and Carter, 1997; Hogan and Frederking, 1998; Bangalore et al., 2001; Jayaraman and Lavie, 2005).

Although statistical MT approaches currently come out as winners in most comparative evaluations, it is clear that the achievable quality of methods relying purely on lookup of fixed phrases will be limited by the simple fact that for any given combination of topic, application scenario, language pair, and text style there will never be sufficient amounts of pre-existing translations to satisfy the needs of purely data-driven approaches.

Rule-based approaches can exploit the effort that goes into single entries in their knowledge repositories in a broader way, as these entries can be unfolded, via rule applications, into large numbers of possible usages. However, this increased generality comes at significant costs for the acquisition of the required knowledge, which needs to be encoded by specialists in formalisms requiring extensive training to be used. In order to push the limits of today's MT technology, integrative approaches will have to be developed that combine the relative advantages of

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 155–158,
Ann Arbor, June 2005.

both paradigms and use them to compensate for their disadvantages. In particular, it should be possible to turn single instances of words and constructions found in training data into internal representations that allow them to be used in more general ways.

In a first step towards the development of integrated solutions, we need to investigate the relative strengths and weaknesses of competing systems on the level of the target text, i.e. find out which sentences and which constructions are rendered well by which type of engine. In a second step, such an analysis will then make it possible to take the outcomes of various engines apart and re-assemble from the building blocks new translations that avoid errors made by the individual engines, i.e. to find integrated solutions that improve over the best of the candidates they have been built from. Once this can be done, the third and final step will involve feed back of corrections into the individual systems, such that differences between system behaviour can trigger (potentially after manual resolution of unclear cases) system updates and mutual learning.

In the long term, one would hope to achieve a setup where a group of MT engines can converge to a committee that typically disagrees only in truly difficult cases. In such a committee, remaining dissent between the members would be a symptom of unresolved ambiguity, that would warrant the cost of manual intervention by the fact that the system as a whole can actually learn from the additional evidence. We expect this setup to be particularly effective when existing MT engines have to be ported to new application domains. Here, a rule-based engine would be able to profit from its more generic knowledge during the early stages of the transition and could teach unseen correspondences of known words and phrases to the SMT engine, whereas the SMT system would bring in its abilities to apply known phrase pairs in novel contexts and quickly learn new vocabulary from examples.

2 Collecting Translation Candidates

2.1 Setting up Statistical MT

In the general picture laid out in the preceding section, statistical MT plays an important role for several reasons. On one hand, the construction of a relatively well-performing phrase-based SMT system from a given set of parallel corpora is no more overly difficult, especially if — as in the case in this shared task — word alignments and a decoder are provided. Furthermore, once the second task in our chain will have been surmounted, it will be relatively easy to feed back building blocks of improved translations into the phrase table, which constitutes the central resource of the SMT system Therefore, SMT facilitates experiments aiming at dynamic and interactive adaptation, the results of which should then also be applicable to MT engines that represent knowledge in a more condensed form.

In order to collect material for testing these ideas, we constructed phrase tables for all four languages, following roughly the procedure given in (Koehn, 2004) but deviating in one detail related to the treatment of unaligned words at the beginning or end of the phrases[1]. We used the Pharaoh decoder as described on http://www.statmt.org/wpt05/mt-shared-task/ after normalization of all tables to lower case.

2.2 Using Commercial Engines

As our main interest is in the integration of statistical and rule-based MT, we tried to collect results from "conventional" MT systems that had more or less uniform characteristics across the languages involved. We could not find MT engines supporting all four source languages, and therefore decided to drop Finnish for this part of the experiment. We sent the texts of the other three languages through several incarnations of Systran-based MT Web-services[2] and through an installation of Lernout & Hauspie Power Translator Pro, Version 6.43.[3]

[1] We used slightly more restrictive conditions that resulted in a 5.76% reduction of phrase table size

[2] The results were incomplete and different, but sufficiently close to each other so that it did not seem worthwhile to explore the differences systematically. Instead we ranked the services according to errors in an informal comparison and took for each sentence the first available translation in this order.

[3] After having collected or computed all translations, we observed that in the case of French, both systems were quite sensitive to the fact that the apostrophes were formatted as separate tokens in the source texts (l ' homme instead of l'homme). We therefore modified and retranslated the French texts, but did not explore possible effects of similar transformations in the other languages.

3 Heuristic Selection

3.1 Approach

We implemented two different ways to select, out of a set of alternative translations of a given sentence, one that looks most promising. The first approach is purely heuristic and is limited to the case where more than two candidates are given. For each candidate, we collect a set of features, consisting of words and word n-grams ($n \in \{2, 3, 4\}$). Each of these features is weighted by the number of candidates it appears in, and the candidate with the largest feature weight per word is taken. This can be seen as the similarity of each of the candidate to a prototypical version composed as a weighted mixture of the collection, or as being remotely related to a sentence-specific language model derived from the candidates. The heuristic measure was used to select "favorite" from each group of competing translations obtained from the same source sentence, yielding a fourth set of translations for the sentences given in DE, FR, and ES.

A particularity of the shared task is the fact that the source sentences of the development and test sets form a parallel corpus. Therefore, we can not only integrate multiple translations of the same source sentence into a hopefully better version, but we can merge the translations of corresponding parts from different source languages into a target form that combines their advantages. This approach, called triangulation in (Kay, 1997), can be motivated by the fact that most cases of translation for dissemination involve multiple target languages; hence one can assume that, except for the very first of them, renderings in multiple languages exist and can be used as input to the next step[4]. See also (Och and Ney, 2001) for some related empirical evidence. In order to obtain a first impression of the potential of triangulation in the domain of parliament debates, we applied the selection heuristics to a set of four translations, one from Finnish, the other three the result of the selections mentioned above.

3.2 Results and Discussion

The BLEU scores (Papineni et al., 2002) for 10 direct translations and 4 sets of heuristic selections thereof are given in Table 1. These results show that in each group of translations for a given source language, the statistical engine came out best. Furthermore, our heuristic approach for the selection of the best among a small set of candidate translations did not result in an increase of the measured BLEU score, but typically gave a score that was only slightly better than the second best of the ingredients. This somewhat disappointing result can be explained in two ways. Apparently, the selection heuristic does not give effective estimates of translation quality for the candidates. Furthermore, the granularity on which the choices have to bee made is too coarse, i.e. the pieces for which the symbolic engines do produce better translations than the SMT engine are accompanied by too many bad choices so that the net effect is negative.

Source Language	MT Engine	BLEU score
DE	Pharaoh	20.48
	L & H	13.97
	Systran	14.92
	heuristic selection	16.01
	statistical selection	20.55
FR	Pharaoh	26.29
	L & H	17.82
	Systran	20.29
	heuristic selection	21.44
	statistical selection	26.49
ES	Pharaoh	26.69
	L & H	17.28
	Systran	17.38
	heuristic selection	19.16
	statistical selection	26.74
FI	Pharaoh	16.76
all	heuristic selection	22.83
	statistical selection	25.80

Table 1: BLEU scores of various MT engines and combinations

4 Statistical Selection

The other score we used was based on probabilities as computed by the trigram language model for English provided by the organizers of the task, in a representation compatible with the SRI LM toolkit

[4] Admittedly, in typical instances of such chains, English would appear earlier.

(Stolcke, 2002). However, a correct implementation for obtaining these estimates was not available in time, so the selections generated from the statistical language model could not be used for official submissions, but were generated and evaluated after the closing date. The results, also displayed in Table 1, show that this approach can lead to slight improvements of the BLEU score, which however turn out not to be statistically sigificant in then sense of (Zhang et al., 2004).

5 Next Steps

When we started the experiments reported here, the hope was to find relatively simple methods to select the best among a small set of candidate translations and to achieve significant improvements of a hybrid architecture over a purely statistical approach. Although we could indeed measure certain improvements, these are not yet big enough for a conclusive "proof of concept". We have started a refinement of our approach that can not only pick the best among translations of complete sentences, but also judge the quality of the building blocks from which the translations are composed. First informal results look very promising. Once we can replace single phrases that appear in one translation by better alternatives taken from a competing candidate, chances are good that a significant increase of the overall translation quality can be achieved.

6 Acknowledgements

This work has been funded by the Deutsche Forschungsgemeinschaft. We want to thank two anonymous reviewers for numerous pointers to relevant literature, Bogdan Sacaleanu for his help with the collection of translations from on-line MT engines, as well as the organizers of the shared task for making these interesting experiments possible.

References

Yasuhiro Akiba, Kenji Imamura, and Eiichiro Sumita. 2001. Using multiple edit distances to automatically rank machine translation output. In *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain.

Yasuhiro Akiba, Taro Watanabe, and Eiichiro Sumita. 2002. Using language and translation models to select the best among outputs from multiple mt systems. In *COLING*.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *ASRU*, Italy.

Chris Callison-Burch and Raymond S. Flournoy. 2001. A program for automatically selecting the best output from multiple machine translation engines. In *Proc. of MT Summit VIII*, Santiago de Compostela, Spain.

Robert E. Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *ANLP*, pages 95–100.

Christopher Hogan and Robert E. Frederking. 1998. An evaluation of the multi-engine mt architecture. In *Proceedings of AMTA*, pages 113–123.

Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. of EAMT*, Budapest, Hungary.

Martin Kay. 1997. The proper place of men and machines in language translation. *Machine Translation*, 12:3–23. First appeared as a Xerox PARC working paper in 1980.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124.

Tadashi Nomoto. 2004. Multi-engine machine translation with voted language model. In *Proc. of ACL*.

Franz-Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain, September.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Manny Rayner and David M. Carter. 1997. Hybrid language processing in the spoken language translator. In *Proc. ICASSP '97*, pages 107–110, Munich, Germany.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*.

Dan Tidhar and Uwe Küssner. 2000. Learning to select a good translation. In *COLING*, pages 843–849.

Wolfgang Wahlster. 2001. Robust translation of spontaneous speech: A multi-engine approach. In *IJCAI*, pages 1484–1493. Invited Talk.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of LREC*, Lisbon, Portugal.

Competitive Grouping in Integrated Phrase Segmentation and Alignment Model

Ying Zhang **Stephan Vogel**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{joy+,vogel+}@cs.cmu.edu

Abstract

This article describes the competitive grouping algorithm at the core of our Integrated Segmentation and Alignment (ISA) model. ISA extracts phrase pairs from a bilingual corpus without requiring the pre-calculated word alignment as many other phrase alignment models do. Experiments conducted within the WPT-05 shared task on statistical machine translation demonstrate the simplicity and effectiveness of this approach.

1 Introduction

In recent years, various phrase translation approaches (Marcu and Wong, 2002; Och et al., 1999; Koehn et al., 2003) have been shown to outperform word-to-word translation models (Brown et al., 1993). Many of these phrase alignment strategies rely on the pre-calculated word alignment and use different heuristics to extract the phrase pairs from the Viterbi word alignment path. The Integrated Segmentation and Alignment (ISA) model (Zhang et al., 2003) does not require such word alignment. ISA segments the sentence into phrases and finds their alignment simultaneously. ISA is simple and fast. Translation experiments have shown comparable performance to other phrase alignment strategies which require complicated statistical model training. In this paper, we describe the key idea behind this model and connect it with the competitive linking algorithm (Melamed, 1997) which was developed for word-to-word alignment.

2 Translation Likelihood as a Statistical Test

Given a bilingual corpus of language pair *F* (Foreign, source language) and *E* (English, target language), if we know the word alignment for each sentence pair we can calculate the co-occurrence frequency for each source/target word pair type $C(f, e)$ and the marginal frequency $C(f) = \sum_e C(f, e)$ and $C(e) = \sum_f C(f, e)$. We can apply various statistical tests (Manning and Schütze, 1999) to measure how likely is the association between f and e, in other words how likely they are mutual translations. In the following sections, we will use χ^2 statistics to measure the the mutual translation likelihood (Church and Hanks, 1990).

3 The Core of the Integrated Phrase Segmentation and Alignment

The competitive linking algorithm (CLA) (Melamed, 1997) is a greedy word alignment algorithm. It was designed to overcome the problem of indirect associations using a simple heuristic: whenever several word tokens f_i in one half of the bilingual corpus co-occur with a particular word token e in the other half of the corpus, the word that is most likely to be e's translation is the one for which the likelihood $L(f, e)$ of translational equivalence is highest. The simplicity of this algorithm depends on a one-to-one alignment assumption. Each word translates to at most one other word. Thus when one pair $\{f, e\}$ is "linked", neither f nor e can be aligned with any other words. This assumption renders CLA unusable in phrase level alignment.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 159–162,
Ann Arbor, June 2005.

We propose an extension, the competitive grouping, as the core component in the ISA model.

3.1 Competitive Grouping Algorithm (CGA)

The key modification to the competitive linking algorithm is to make it less greedy. When a word pair is found to be the winner of the competition, we allow it to invite its neighbors to join the "winner's club" and group them together as an aligned phrase pair. The one-to-one assumption is thus discarded in CGA. In addition, we introduce the *locality* assumption for phrase alignment. *Locality* states that a source phrase of adjacent words can only be aligned to a target phrase composed of adjacent words. This is not true of most language pairs in cases such as the relative clause, passive tense, and prepositional clause, etc.; however this assumption renders the problem tractable. Here is a description of CGA:

For a sentence pair $\{\mathbf{f}, \mathbf{e}\}$, represent the word pair statistics for each word pair $\{f, e\}$ in a two dimensional matrix $L_{I\times J}$, where $L(i,j) = \chi^2(f_i, e_j)$ in our implementation. [1]

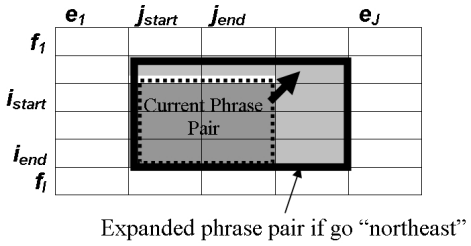


Figure 1: Expanding the current phrase pair

Denote an aligned phrase pair $\{\tilde{f}, \tilde{e}\}$ as a tuple $[i_{start}, i_{end}, j_{start}, j_{end}]$ where $\tilde{f}$ is $f_{i_{start}}, f_{i_{start+1}}, \ldots, f_{i_{end}}$ and similarly for $\tilde{e}$.

1. Find i^* and j^* such that $L(i^*, j^*)$ is the highest. Create a *seed* phrase pair $[i^*, i^*, j^*, j^*]$ which is simply the word pair $\{f_{i^*}, e_{j^*}\}$ itself.

2. Expand the current phrase pair $[i_{start}, i_{end}, j_{start}, j_{end}]$ to the neighboring territory to include adjacent source and target words in the phrase alignment group. There are 8 ways to group new words into the phrase pair. For example, one can expand to the north by including an additional source word $f_{i_{start}-1}$ to be aligned with all the target words in the current group; or one can expand to the northeast by including $f_{i_{start}-1}$ and $e_{j_{end}+1}$ (Figure 1).

   Two criteria have to be satisfied for each expansion:

   (a) If a new source word $f_{i'}$ is to be grouped, $\max_{j_{start}\le j\le j_{end}} L(i', j)$ should be no smaller than $\max_{1\le j\le J} L(i', j)$. Since CGA is a greedy algorithm as described below, this is to guarantee that $f_{i'}$ will not "regret" the decision of joining the phrase pair because it does not have other "better" target words to be aligned with. Similar constraint is applied if a new target word $e_{j'}$ is to be grouped.

   (b) The highest value in the newly-expanded area needs to be "similar" to the seed value $L(i^*, j^*)$.

   Expand the current phrase pair to the largest extend possible as long as both criteria are satisfied.

3. The locality assumption means that the aligned phrase cannot be aligned again. Therefore, all the source and target words in the phrase pair are marked as "invalid" and will be skipped in the following steps.

4. If there is another valid pair $\{f_i, e_j\}$, then repeat from Step 1.

Figure 2 and Figure 3 show a simple example of applying CGA on the sentence pair {*je déclare reprise la session/i declare resumed the session*}.

	i	declare	resumed	the	session
je	40316.90	0.79	0.01	19.39	0.04
déclare	0.40	760.79	40.85	0.33	86.78
reprise	0.01	24.66	312.73	0.31	402.86
la	10.50	0.01	0.17	667.49	1.60
session	0.00	40.42	5.13	0.80	3795.00

Figure 2: Seed pair {*je / i*}, no expansion allowed

[1] χ^2 statistics were found to be more discriminative in our experiments than other symmetric word association measures, such as the averaged mutual information, ϕ^2 statistics and Dice-coefficient.

	i	declare	resumed	the	session
je					
déclare		760.79	40.85	0.33	86.78
reprise		24.66	312.73	0.31	402.86
la		0.01	0.17	667.49	1.60
session		40.42	5.13	0.80	3795.00

Figure 3: Seed pair {*session/session*}, expanded to {*la session/the session*}

3.2 Exploring all possible groupings

The similarity criterion 2-(b) described previously is used to control the granularity of phrase pairs. In cases where the pairs $\{f_1 f_2, e_1 e_2\}$, $\{f_1, e_1\}$ and $\{f_2, e_2\}$ are all valid translations pairs, similarity is used to control whether we want to align $\{f_1 f_2, e_1 e_2\}$ as one phrase pair or two shorter ones.

The granularity of the phrase pairs is hard to optimize especially when the test data is unknown. On the one hand, we prefer long phrases since interaction among the words in the phrase, for example word sense, morphology and local reordering could be encapsulated. On the other hand, long phrase pairs are less likely to occur in the test data than the shorter ones and may lead to low coverage. To have both long and short phrases in the alignment, we apply a range of similarity thresholds for each of the expansion operations. By applying a low similarity threshold, the expanded phrase pairs tend to be large, while a higher similarity threshold results in shorter phrase pairs. As described above, CGA is a greedy algorithm and the expansion of the seed pair restricts the possible alignments for the rest of the sentence. Figure 4 shows an example as we explore all the possible grouping choices in a depth-first search. In the end, all unique phrase pairs along the path traveled are output as phrase translation candidates for the current sentence pair.

3.3 Phrase translation probabilities

Each aligned phrase pair $\{\tilde{f}, \tilde{e}\}$ is assigned a likelihood score $L(\tilde{f}, \tilde{e})$, defined as:

$$\frac{\sum_i \max_j \log L(f_i, e_j) + \sum_j \max_i \log L(f_i, e_j)}{|\tilde{f}| + |\tilde{e}|}$$

where i ranges over all words in $\tilde{f}$ and similarly j in $\tilde{e}$.

Given the collected phrase pairs and their likelihood, we estimate the phrase translation probability

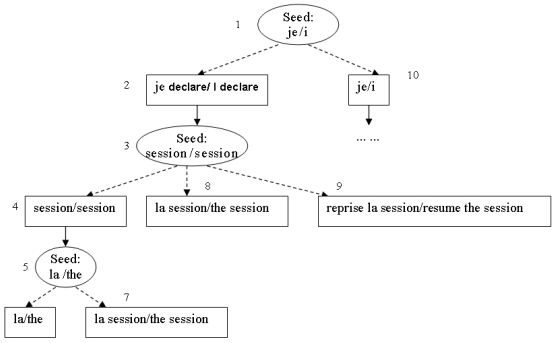


Figure 4: Depth-first itinerary of all possible grouping choices.

by their weighted frequency:

$$P(\tilde{f}|\tilde{e}) = \frac{count(\tilde{f}, \tilde{e}) \cdot L(\tilde{f}, \tilde{e})}{\sum_{\tilde{f}} count(\tilde{f}, \tilde{e}) \cdot L(\tilde{f}, \tilde{e})}$$

No smoothing is applied to the probabilities.

4 Learning co-occurrence information

In most cases, word alignment information is not given and is treated as a hidden parameter in the training process. We initialize a word pair co-occurrence frequency by assuming uniform alignment for each sentence pair, i.e. for sentence pair $(\mathbf{f}, \mathbf{e})$ where $\mathbf{f}$ has I words and $\mathbf{e}$ has J words, each word pair $\{f, e\}$ is considered to be aligned with frequency $\frac{1}{I \times J}$. These co-occurrence frequencies will be accumulated over the whole corpus to calculate the initial $L(f, e)$. Then we iterate the ISA model:

1. Apply the competitive grouping algorithm to each sentence pair to find all possible phrase pairs.
2. For each identified phrase pair $\{\tilde{f}, \tilde{e}\}$, increase the co-occurrence counts for all word pairs inside $\{\tilde{f}, \tilde{e}\}$ with weight $\frac{1}{|\tilde{f}| \cdot |\tilde{e}|}$.
3. Calculate $L(f, e)$ again and goto Step 1 for several iterations.

5 Experiments

We participated the shared task in the WPT05 workshop[2] and applied ISA to all four language pairs

[2] http://www.statmt.org/wpt05/mt-shared-task/

(French-English, Finnish-English, German-English and Spanish-English). Table 1 shows the n-gram coverage of the dev-test set. French and Spanish data are better covered by the training data compared to the German and Finnish sets. Since our phrase alignment is constrained by the locality assumption and we can only extract phrase pairs of adjacent words, lower n-gram coverage will result in lower translation scores. We used the training data

Dev-test	DE	ES	FI	FR
N=1	99.2	99.6	98.2	99.8
N=2	88.2	93.3	73.0	94.7
N=3	59.4	71.7	38.2	76.0
N=4	30.0	42.9	17.0	50.6
N=5	13.0	21.7	6.8	29.8
N=16	(8)	(65)	(1)	(101)
N=19	(1)	(23)		(34)
N=23		(1)		(1)

Table 1: Percentage of dev-test n-grams covered by the training data. Numbers in parenthesis are the actual counts of n-gram tokens in the dev-test data.

and the language model as provided and manually tuned the parameters of the Pharaoh decoder[3] to optimize BLEU scores. Table 2 shows the translation results on the dev-test and the test set of WPT05. The BLEU scores appear comparable to those of other state-of-the-art phrase alignment systems, in spite of the simplicity of the ISA model and ease of training.

	DE	ES	FI	FR
Dev-test	18.63	26.20	12.88	26.20
Test	18.93	26.14	12.66	26.71

Table 2: BLEU scores of ISA in WPT05

6 Conclusion

In this paper, we introduced the competitive grouping algorithm which is at the core of the ISA phrase alignment model. As an extension to the competitive linking algorithm which is used for word-to-word alignment, CGA overcomes the assumption of one-to-one mapping and makes it possible to align phrase pairs. Despite its simplicity, the ISA model has achieved competitive translation results. We plan to release ISA toolkit[4] to the community in the near future.

[3] http://www.isi.edu/licensed-sw/pharaoh/

[4] http://projectile.is.cs.cmu.edu/research/public/isa/index.htm

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadephia, PA, July 6-7.

I. Dan Melamed. 1997. A word-to-word model of translational equivalence. In *Proceedings of the 8-th conference on EACL*, pages 490–497, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2003. Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proceedings of NLP-KE'03*, Beijing, China, October.

Deploying Part-of-Speech Patterns to Enhance Statistical Phrase-Based Machine Translation Resources

Christina Lioma
Department of Computing Science
University of Glasgow
G12 8QQ
xristina@dcs.gla.ac.uk

Iadh Ounis
Department of Computing Science
University of Glasgow
G12 8QQ
ounis@dcs.gla.ac.uk

Abstract

Part-of-Speech patterns extracted from parallel corpora have been used to enhance a translation resource for statistical phrase-based machine translation.

1 Introduction

The use of structural and syntactic information in language processing implementations in recent years has been producing contradictory results. Whereas language generation has benefited from syntax [Wu, 1997; Alshawi et al., 2000], the performance of statistical phrase-based machine translation when relying solely on syntactic phrases has been reported to be poor [Koehn et al., 2003].

We carry out a set of experiments to explore whether heuristic learning of part-of-speech patterns from a parallel corpus can be used to enhance phrase-based translation resources.

2 System

The resources used for our experiments are as follows. The statistical machine translation GIZA++ toolkit was used to generate a bilingual translation table from the French-English parallel and sentence-aligned Europarl corpus. Additionally, a phrase table generated from the Europarl French-English corpus, and a training test set of 2000 French and English sentences that were made available on the webpage of the ACL 2005 workshop[1] were also used. Syntactic tagging was realized by the TreeTagger, which is a probabilistic part-of-speech tagger and lemmatizer. The decoder used to produce machine translations was Pharaoh, version 1.2.3.

We used GIZA++ to generate a translation table from the parallel corpus. The table produced consisted of individual words and phrases, followed by their corresponding translation and a unique probability value. Specifically, every line of the said table consisted of a French entry (in the form of one or more tokens), followed by an English entry (in the form of one or more tokens), followed by $P(f|e)$, which is the probability P of translation to the French entry f given the English entry e. We added the GIZA++-generated table to the phrase-based translation table downloaded from the workshop webpage. During this merging of translation tables, no word or phrase was omitted, replaced or altered. We chose to combine the two aforementioned translation tables in order to achieve better coverage. We called the resulting merged translation table *lexical phrase table*.

In order to utilize the syntactic information stemming from our resources, we used the TreeTagger to tag both the parallel corpus and the *lexical phrase table*. The probability values included in the *lexical phrase table* were not tagged. The TreeTagger uses a slightly modified version of the Penn Treebank tagset, different for each language. In order to achieve tag-uniformity, we performed the following dual tag-smoothing operation.

[1] The Europarl French-English corpus and phrase table, and the training test set are available at: http://www.statmt.org/wpt05/mt-shared-task/

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 163–166,
Ann Arbor, June 2005.

Firstly, we changed the French tags into their English equivalents, i.e. NOM (noun – French) became NN (noun – English). Secondly, we simplified the tags, so that they reflected nothing more than general part-of-speech information. For example, tags denoting predicate-argument structures, wh-movement, passive voice, inflectional variation, and so on, were simplified. For example, NNS (noun – plural) became NN (noun).

Once our resources were uniformly tagged, we used them to extract part-of-speech correspondences between the two languages. Specifically, we extracted a sentence-aligned parallel corpus of French and English part-of-speech patterns from the tagged Europarl parallel corpus. We called this corpus of parallel and corresponding part-of-speech patterns *pos-corpus*. The format of the *pos-corpus* remained identical to the format of the original parallel corpus, with the sole difference that individual words were replaced by their corresponding part-of-speech tag. Similarly, we extracted a translation table of part-of-speech patterns from the tagged *lexical phrase table*. We called this part-of-speech translation table *pos-table*. The *pos-table* had exactly the same format as *the lexical phrase table*, with the unique difference that individual words were replaced by their corresponding part-of-speech tag. The translation probability values included in the *lexical phrase table* were copied onto the *pos-table* intact.

Each of the part-of-speech patterns contained in the *pos-corpus* was matched against the part-of-speech patterns contained in the *pos-table*. Matching was realized similarly to conventional left-to-right string matching operations. Matching was considered to be successful not simply when a part-of-speech pattern was found to be contained in, or part of a longer pattern, but when patterns were found to be absolutely identical. When a perfect match was found, the translation probability value of the specific pattern in the *pos-table* was increased to the maximum value of 1. If the score were already 1, it remained unchanged. When there were no matches, values remained unchanged. We chose to match identical part-of-speech patterns, and not to accept partial pattern matches, because the latter would require a revision of our probability recomputation method. This point is discussed in section 3 of this paper.

Once all matching was complete, the newly enhanced *pos-table*, which now contained translation probability scores reflecting the syntactic features of the relevant languages, was used to update the original *lexical phrase table*. This update consisted in matching each and every part-of-speech pattern with its original lexical phrase, and replacing the initial translation probability score with the values contained in the *pos-table*. The identification of the original lexical phrases that generated each and every part-of-speech pattern was facilitated by the use of pattern-identifiers (*pos-ids*) and phrase-identifiers (*phrase-ids*), which were introduced at a very early stage in the process for that purpose. The resulting translation phrase table contained exactly the same entries as the *lexical phrase table*, but had different probability scores assigned to some of these entries, in line with the parallel part-of-speech co-occurrences and correspondences found in the Europarl corpus. We called this table *enhanced phrase table*. Table 1 illustrates the process described above with the example of a phrase, the part-of-speech analysis of which has been used to increase its original translation probability value from 0.333333 to 1.

Lexical phrase table
actions extérieures | external action | 0.333333
Tagged *lexical phrase table*
actions_NN extérieures_JJ | external_JJ action_NN | 0.333333
pos-corpus
NN JJ | JJ NN
Enhanced phrase table
actions extérieures | external action | 1

Table 1: Extracting and matching a part-of-speech pattern to increase translation probability.

We used the Pharaoh decoder firstly with our *lexical phrase table*, and secondly with our *enhanced phrase table* in order to generate statistical machine translations of source and target language variations of the French and English training test set. We measured performance using the BLEU score [Papineri et al., 2001], which estimates the accuracy of translation output with respect to a reference translation. For both source-target language combinations, the use of the *lexical phrase table* received a slightly lower score than the score achieved when using the *enhanced phrase table*. The difference between these two approaches is not significant (p-value > 0.05). The results of our

experiments are displayed in Table 2 and discussed in Section 3.

Language Pair	Lexical	Enhanced
English-French	25.50	25.63
French-English	26.59	26.89

Table 2: Our translation performance (measured with BLEU)

3 Discussion

The motivation behind this investigation has been to test whether syntactic or structural language aspects can be reflected or represented in the resources used in statistical phrase-based machine translation.

We adopted a line of investigation that concentrates on the correspondence of part-of-speech patterns between French and English. We measured the usability of syntactic structures for statistical phrase-based machine translation by comparing translation performance when a standard phrase table was used, and when a syntactically enhanced phrase table was used. Both approaches scored very similarly. This similarity in the performance is justified by the following three factors.

Firstly, the difference between the two translation resources, namely the *lexical phrase table* and the *enhanced phrase table*, does not relate to their entries, and thus their coverage, but to a simple alteration of the translation probability values of some of their entries. The coverage of these resources is exactly identical.

Secondly, a closer examination of the translation probability value alterations that took place in order to reflect part-of-speech correspondences reveals that the proportion of the entries of the phrase table that were matched syntactically to phrases from the parallel corpus, and thus underwent a modification in their translation probability score, was very low (less than 1%). The reason behind this is the fact that the part-of-speech patterns produced by the parallel corpus were long strings in their vast majority, while the part-of-speech patterns found in the phrase table were significantly shorter strings. The inclusion of phrases longer than three words in translation resources has been avoided, as it has been shown not to have a strong impact on translation performance [Koehn et al., 2003].

Thirdly, the above described translation probability value modifications were not parameterized, but consisted in a straightforward increase of the translation probability to its maximum value. It remains to be seen how these probability value alterations can be expanded to a type of probability value 'reweighing', in line with specific parameters, such as the size of the resources involved, the frequency of part-of-speech patterns in the resources, the length of part-of-speech patterns, as well as the syntactic classification of the members of part-of-speech patterns. If one is to compare the impact that such parameters have had upon the performance of automatic information summarisation [Mani, 2001] and retrieval technology [Belew, 2000], it may be worth experimenting with such parameter tuning when refining machine translation resources.

A note should be made to the choice of tagger for our experiments. A possible risk when attempting any syntactic examination of a large set of data may stem from the overriding role that syntax often assumes over semantics. Statistical phrase-based machine translation has been faced with instances of this phenomenon, often disguised as linguistic idiosyncrasies. This phenomenon accounts for such instances as when nouns appear in pronominal positions, or as adverbial modifiers. On these occasions, and in order for the syntactic examination to be precise, words would have to be defined on the basis of their syntactic distribution rather than their semantic function. The TreeTagger abides by this convention, which is one of the main reasons why we chose it over a plethora of other freely available taggers, the remaining reasons being its high speed and low error rate. In addition, it should be clarified that there is no statistical, linguistic, or other reason why we chose to adopt the English version of the Penn TreeBank tagset over the French, as they are both equally conclusive and transparent.

The overall driving force behind our investigation has been to test whether part-of-speech structures can be of assistance to the enhancement of translation resources for statistical phrase-based machine translation. We view our use of part-of-speech patterns as a natural extension to the introduction of structural elements to statistical machine translation by Wang [1998] and Och et al. [1999].

Our empirical results suggest that the use of part-of-speech pattern correspondences to enhance existing translation resources does not damage machine translation performance. What remains to be investigated is how this approach can be optimized, and how it would respond to known statistical machine translation issues, such as mapping nested structures, or the handling of 'unorthodox' language pairs, i.e. agglutinative-fusion languages.

4 Conclusion

Syntactic and structural language information contained in a bilingual parallel corpus has been extracted and used to refine the translation probability values of a translation phrase table, using simple heuristics. The usability of the said translation table in statistical phrase-based machine translation has been tested in the shared task of the second track of the ACL 2005 Workshop on Building and Using Parallel Corpora. Findings suggest that using part-of-speech information to alter translation probabilities has had no significant effect upon translation performance. Further investigation is required to reveal how our approach can be optimized in order to produce significant performance improvement.

References

Alshawi, H., Bangalore, S., and Douglas, S. (2000). Learning Dependency Translation Models as Collections of Finite State Head Transducers. *Computational Linguistics*, 26(1).

Belew, R. K. (2000). *Finding Out About: Search Engine Technology from a Cognitive Perspective*. Cambridge University Press, USA.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology Conference 2003 (HLT/NAACL 2003)*, pages 127-133.

Mani, I. (2001). *Automatic Summarization*. John Benjamins Publishing Company, Amsterdam.

Och, F. J., Tilmann, C., and Ney, H. (1999). Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint SIGDAT Conference of Empirical Methods in Natural Language Processing and Very Large Corpora 1999 (EMNLP 1999)*, pages 20-28.

Papineri, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: A Method for Automatic Evaluation of Machine Translation. Technical Report RC22176(W0109-022), IBM Research Report.

Wang, Y. (1998). Grammar Inference and Statistical Machine Translation. Ph.D. thesis, Carnegie Melon University.

Wu, D. (1997). Stochastic Inversion transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3).

Yamada, K. and Knight, K. (2001). A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 39)*, pages 6-11.

Novel Reordering Approaches in Phrase-Based Statistical Machine Translation

Stephan Kanthak, David Vilar, Evgeny Matusov, Richard Zens, and Hermann Ney
The authors are with the Lehrstuhl für Informatik VI,
Computer Science Department, RWTH Aachen University,
D-52056 Aachen, Germany.
E-mail: {kanthak,vilar,matusov,zens,ney}@informatik.rwth-aachen.de.

Abstract

This paper presents novel approaches to reordering in phrase-based statistical machine translation. We perform consistent reordering of source sentences in training and estimate a statistical translation model. Using this model, we follow a phrase-based monotonic machine translation approach, for which we develop an efficient and flexible reordering framework that allows to easily introduce different reordering constraints. In translation, we apply source sentence reordering on word level and use a reordering automaton as input. We show how to compute reordering automata on-demand using IBM or ITG constraints, and also introduce two new types of reordering constraints. We further add weights to the reordering automata. We present detailed experimental results and show that reordering significantly improves translation quality.

1 Introduction

Reordering is of crucial importance for machine translation. Already (Knight et al., 1998) use full unweighted permutations on the level of source words in their early weighted finite-state transducer approach which implemented single-word based translation using conditional probabilities. In a refinement with additional phrase-based models, (Kumar et al., 2003) define a probability distribution over all possible permutations of source sentence phrases and prune the resulting automaton to reduce complexity.

A second category of finite-state translation approaches uses joint instead of conditional probabilities. Many joint probability approaches originate in speech-to-speech translation as they are the natural choice in combination with speech recognition models. The automated transducer inference techniques OMEGA (Vilar, 2000) and GIATI (Casacuberta et al., 2004) work on phrase level, but ignore the reordering problem from the view of the model. Without reordering both in training and during search, sentences can only be translated properly into a language with similar word order. In (Bangalore et al., 2000) weighted reordering has been applied to target sentences since defining a permutation model on the source side is impractical in combination with speech recognition. In order to reduce the computational complexity, this approach considers only a set of plausible reorderings seen on training data.

Most other phrase-based statistical approaches like the Alignment Template system of Bender et al. (2004) rely on (local) reorderings which are implicitly memorized with each pair of source and target phrases in training. Additional reorderings on phrase level are fully integrated into the decoding process, which increases the complexity of the system and makes it hard to modify. Zens et al. (2003) reviewed two types of reordering constraints for this type of translation systems.

In our work we follow a phrase-based translation approach, applying source sentence reordering on word level. We compute a reordering graph on-demand and take it as input for monotonic translation. This approach is modular and allows easy introduction of different reordering constraints and probabilistic dependencies. We will show that it performs at least as well as the best statistical machine translation system at the IWSLT Evaluation.

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 167–174,
Ann Arbor, June 2005.

In the next section we briefly review the basic theory of our translation system based on weighted finite-state transducers (WFST). In Sec. 3 we introduce new methods for reordering and alignment monotonization in training. To compare different reordering constraints used in the translation search process we develop an on-demand computable framework for permutation models in Sec. 4. In the same section we also define and analyze unrestricted and restricted permutations with some of them being first published in this paper. We conclude the paper by presenting and discussing a rich set of experimental results.

2 Machine Translation using WFSTs

Let f_1^J and e_i^I be two sentences from a source and target language. Assume that we have word level alignments $\mathcal{A}$ of all sentence pairs from a bilingual training corpus. We denote with $\tilde{e}_1^J$ the segmentation of a target sentence e_1^I into J phrases such that f_1^J and $\tilde{e}_1^J$ can be aligned to form bilingual tuples $(f_j, \tilde{e}_j)$. If alignments are only *functions of target words* $A' : \{1, \ldots, I\} \rightarrow \{1, \ldots, J\}$, the bilingual tuples $(f_j, \tilde{e}_j)$ can be inferred with e. g. the GIATI method of (Casacuberta et al., 2004), or with our novel monotonization technique (see Sec. 3). Each source word will be mapped to a target phrase of one or more words or an "empty" phrase ε. In particular, the source words which will remain non-aligned due to the alignment functionality restriction are paired with the empty phrase.

We can then formulate the problem of finding the best translation $\hat{e}_1^I$ of a source sentence f_1^J:

$$\begin{aligned}
\hat{e}_1^I &= \operatorname*{argmax}_{e_1^I} Pr(f_1^J, e_1^I) \\
&= \operatorname*{argmax}_{\tilde{e}_1^J} \sum_{A \in \mathcal{A}} Pr(f_1^J, \tilde{e}_1^J, A) \\
&\cong \operatorname*{argmax}_{\tilde{e}_1^J} \max_{A \in \mathcal{A}} Pr(A) \cdot Pr(f_1^J, \tilde{e}_1^J | A) \\
&\cong \operatorname*{argmax}_{\tilde{e}_1^J} \max_{A \in \mathcal{A}} \prod_{f_j : j=1\ldots J} Pr(f_j, \tilde{e}_j | f_1^{j-1}, \tilde{e}_1^{j-1}, A) \\
&= \operatorname*{argmax}_{\tilde{e}_1^J} \max_{A \in \mathcal{A}} \prod_{f_j : j=1\ldots J} p(f_j, \tilde{e}_j | f_{j-m}^{j-1}, \tilde{e}_{j-m}^{j-1}, A)
\end{aligned}$$

In other words: if we assume a uniform distribution for $Pr(A)$, the translation problem can be mapped to the problem of estimating an m-gram language model over a learned set of bilingual tuples $(f_j, \tilde{e}_j)$. Mapping the bilingual language model to a WFST T is canonical and it has been shown in (Kanthak et al., 2004) that the search problem can then be rewritten using finite-state terminology:

$$\hat{e}_1^I = \texttt{project-output}(\texttt{best}(f_1^J \circ T)) .$$

This implementation of the problem as WFSTs may be used to efficiently solve the search problem in machine translation.

3 Reordering in Training

When the alignment function A' is not monotonic, target language phrases $\tilde{e}$ can become very long. For example in a completely non-monotonic alignment all target words are paired with the last aligned source word, whereas all other source words form tuples with the empty phrase. Therefore, for language pairs with big differences in word order, probability estimates may be poor.

This problem can be solved by reordering either source or target training sentences such that alignments become monotonic for all sentences. We suggest the following consistent source sentence reordering and alignment monotonization approach in which we compute optimal, minimum-cost alignments.

First, we estimate a cost matrix C for each sentence pair (f_1^J, e_1^I). The elements of this matrix c_{ij} are the local costs of aligning a source word f_j to a target word e_i. Following (Matusov et al., 2004), we compute these local costs by interpolating state occupation probabilities from the source-to-target and target-to-source training of the HMM and IBM-4 models as trained by the GIZA++ toolkit (Och et al., 2003). For a given alignment $A \subseteq I \times J$, we define the costs of this alignment $c(A)$ as the sum of the local costs of all aligned word pairs:

$$c(A) = \sum_{(i,j) \in A} c_{ij} \qquad (1)$$

The goal is to find an alignment with the minimum costs which fulfills certain constraints.

3.1 Source Sentence Reordering

To reorder a source sentence, we require the alignment to be a *function* of *source* words A_1: $\{1, \ldots, J\} \rightarrow \{1, \ldots, I\}$, easily computed from the cost matrix C as:

$$A_1(j) = \operatorname{argmin}_i c_{ij} \qquad (2)$$

We do not allow for non-aligned source words. A_1 naturally defines a new order of the source words f_1^J which we denote by $\breve{f}_1^J$. By computing this permutation for each pair of sentences in training and applying it to each source sentence, we create a corpus of reordered sentences.

3.2 Alignment Monotonization

In order to create a "sentence" of bilingual tuples $(\breve{f}_1^J, \tilde{e}_1^J)$ we required alignments between reordered source and target words to be a *function* of *target* words $A_2 : \{1, \ldots, I\} \rightarrow \{1, \ldots, J\}$. This alignment can be computed in analogy to Eq. 2 as:

$$A_2(i) = \text{argmin}_j \breve{c}_{ij} \quad (3)$$

where $\breve{c}_{ij}$ are the elements of the new cost matrix $\breve{C}$ which corresponds to the reordered source sentence. We can optionally re-estimate this matrix by repeating EM training of state occupation probabilities with GIZA++ using the reordered source corpus and the original target corpus. Alternatively, we can get the cost matrix $\breve{C}$ by reordering the columns of the cost matrix C according to the permutation given by alignment A_1.

In alignment A_2 some target words that were previously unaligned in A_1 (like "the" in Fig. 1) may now still violate the alignment monotonicity. The monotonicity of this alignment can not be guaranteed for *all* words if re-estimation of the cost matrices had been performed using GIZA++.

The general GIATI technique (Casacuberta et al., 2004) is applicable and can be used to monotonize the alignment A_2. However, in our experiments the following method performs better. We make use of the cost matrix representation and compute a monotonic minimum-cost alignment with a dynamic programming algorithm similar to the Levenshtein string edit distance algorithm. As costs of each "edit" operation we consider the local alignment costs. The resulting alignment A_3 represents a minimum-cost monotonic "path" through the cost matrix. To make A_3 a function of target words we do not consider the source words non-aligned in A_2 and also forbid "deletions" ("many-to-one" source word alignments) in the DP search.

An example of such consistent reordering and monotonization is given in Fig. 1. Here, we reorder the German source sentence based on the initial alignment A_1, then compute the function of target words A_2, and monotonize this alignment to A_3

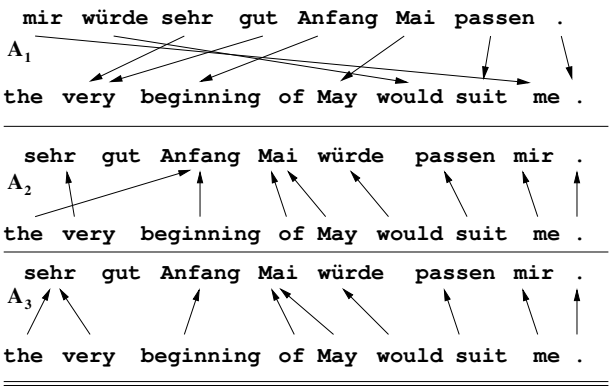


Figure 1: Example of alignment, source sentence reordering, monotonization, and construction of bilingual tuples.

with the dynamic programming algorithm. Fig. 1 also shows the resulting bilingual tuples $(\breve{f}_j, \tilde{e}_j)$.

4 Reordering in Search

When searching the best translation $\tilde{e}_1^J$ for a given source sentence f_1^J, we permute the source sentence as described in (Knight et al., 1998):

$$\hat{e}_1^I = \texttt{project-output}(\texttt{best}(\texttt{permute}(f_1^J) \circ T))$$

Permuting an input sequence of J symbols results in $J!$ possible permutations and representing the permutations as a finite-state automaton requires at least 2^J states. Therefore, we opt for computing the permutation automaton on-demand while applying beam pruning in the search.

4.1 Lazy Permutation Automata

For on-demand computation of an automaton in the flavor described in (Kanthak et al., 2004) it is sufficient to specify a state description and an algorithm that calculates all outgoing arcs of a state from the state description. In our case, each state represents a permutation of a subset of the source words f_1^J, which are already translated.

This can be described by a bit vector b_1^J (Zens et al., 2002). Each bit of the state bit vector corresponds to an arc of the linear input automaton and is set to one if the arc has been used on any path from the initial to the current state. The bit vectors of two states connected by an arc differ only in a single bit. Note that bit vectors elegantly solve the problem of recombining paths in the automaton as states with

the same bit vectors can be merged. As a result, a fully minimized permutation automaton has only a single initial and final state.

Even with on-demand computation, complexity using full permutations is unmanagable for long sentences. We further reduce complexity by additionally constraining permutations. Refer to Figure 2 for visualizations of the permutation constraints which we describe in the following.

4.2 IBM Constraints

The IBM reordering constraints are well-known in the field of machine translation and were first described in (Berger et al., 1996). The idea behind these constraints is to deviate from monotonic translation by postponing translations of a limited number of words. More specifically, at each state we can translate any of the *first* l yet uncovered word positions. The implementation using a bit vector is straightforward. For consistency, we associate window size with the parameter l for all constraints presented here.

4.3 Inverse IBM Constraints

The original IBM constraints are useful for a large number of language pairs where the ability to skip some words reflects the differences in word order between the two languages. For some other pairs, it is beneficial to translate some words at the end of the sentence first and to translate the rest of the sentence nearly monotonically. Following this idea we can define the *inverse IBM constraints*. Let j be the first uncovered position. We can choose any position for translation, unless $l-1$ words on positions $j' > j$ have been translated. If this is the case we must translate the word in position j. The inverse IBM constraints can also be expressed by

$$\texttt{invIBM}(x) = \texttt{transpose}(\texttt{IBM}(\texttt{transpose}(x)))\,.$$

As the `transpose` operation can not be computed on-demand, our specialized implementation uses bit vectors b_1^J similar to the IBM constraints.

4.4 Local Constraints

For some language pairs, e.g. Italian – English, words are moved only a few words to the left or right. The IBM constraints provide too many alternative permutations to chose from as each word can be moved to the end of the sentence. A solution that allows only for local permutations and therefore has very low complexity is given by the following permutation rule: the next word for translation comes from the window of l positions[1] counting from the first yet uncovered position. Note, that the local constraints define a true subset of the permutations defined by the IBM constraints.

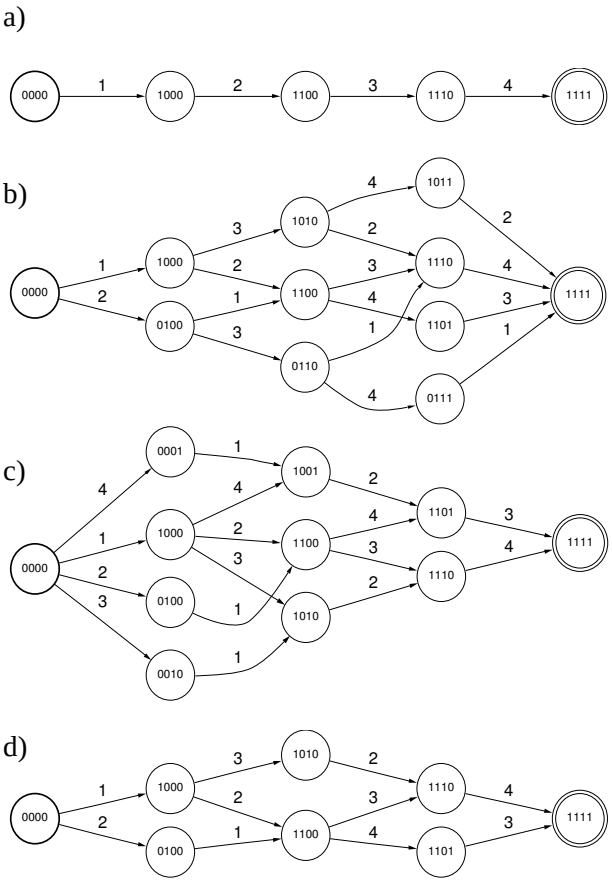


Figure 2: Permutations of a) positions $j = 1, 2, 3, 4$ of a source sentence $f_1 f_2 f_3 f_4$ using a window size of 2 for b) IBM constraints, c) inverse IBM constraints and d) local constraints.

4.5 ITG Constraints

Another type of reordering can be obtained using Inversion Transduction Grammars (ITG) (Wu, 1997). These constraints are inspired by bilingual bracketing. They proved to be quite useful for machine translation, e.g. see (Bender et al., 2004). Here, we interpret the input sentence as a sequence of segments. In the beginning, each word is a segment of its own. Longer segments are constructed by recursively combining two adjacent segments. At each

[1] both covered and uncovered

		Chinese	English	Japanese	English	Italian	English
train	sentences	20 000		20 000		66107	
	words	182 904	160 523	209 012	160 427	410 275	427 402
	singletons	3 525	2 948	4 108	2 956	6 386	3 974
	vocabulary	7 643	6 982	9 277	6 932	15 983	10 971
dev	sentences	506		506		500	
	words	3 515	3 595	4 374	3 595	3 155	3 253
	sentence length (avg/max)	6.95 / 24	7.01 / 29	8.64 / 30	7.01 / 29	5.79 / 24	6.51 / 25
test	sentences	500		500		506	
	words	3 794	–	4 370	–	2 931	3 595
	sentence length (avg/max)	7.59 / 62	7.16 / 71	8.74 / 75	7.16 / 71	6.31 / 27	6.84 / 28

Table 1: Statistics of the Basic Travel Expression (BTEC) corpora.

combination step, we either keep the two segments in monotonic order or invert the order. This process continues until only one segment for the whole sentence remains. The on-demand computation is implemented in spirit of Earley parsing.

We can modify the original ITG constraints to further limit the number of reorderings by forbidding segment inversions which violate IBM constraints with a certain window size. Thus, the resulting reordering graph contains the intersection of the reorderings with IBM and the original ITG constraints.

4.6 Weighted Permutations

So far, we have discussed how to generate the permutation graphs under different constraints, but permutations were equally probable. Especially for the case of nearly monotonic translation it is make sense to restrict the degree of non-monotonicity that we allow when translating a sentence. We propose a simple approach which gives a higher probability to the monotone transitions and penalizes the non-monotonic ones.

A state description b_1^J, for which the following condition holds:

$$Mon(j): \; b_{j'} = \delta(j' \leq j) \;\; \forall \, 1 \leq j' \leq J$$

represents the monotonic path up to the word f_j. At each state we assign the probability α to that outgoing arc where the target state description fullfills $Mon(j+1)$ and distribute the remaining probability mass $1-\alpha$ uniformly among the remaining arcs. In case there is no such arc, all outgoing arcs get the same uniform probability. This weighting scheme clearly depends on the state description and the outgoing arcs only and can be computed on-demand.

5 Experimental Results

5.1 Corpus Statistics

The translation experiments were carried out on the *Basic Travel Expression Corpus* (BTEC), a multilingual speech corpus which contains tourism-related sentences usually found in travel phrase books. We tested our system on the so called Chinese-to-English (CE) and Japanese-to-English (JE) Supplied Tasks, the corpora which were provided during the International Workshop on Spoken Language Translation (IWSLT 2004) (Akiba et al., 2004). In addition, we performed experiments on the Italian-to-English (IE) task, for which a larger corpus was kindly provided to us by ITC/IRST. The corpus statistics for the three BTEC corpora are given in Tab. 1. The development corpus for the Italian-to-English translation had only one reference translation of each Italian sentence. A set of 506 source sentences and 16 reference translations is used as a development corpus for Chinese-to-English and Japanese-to-English and as a test corpus for Italian-to-English tasks. The 500 sentence Chinese and Japanese test sets of the IWSLT 2004 evaluation campaign were translated and automatically scored against 16 reference translations after the end of the campaign using the IWSLT evaluation server.

5.2 Evaluation Criteria

For the automatic evaluation, we used the criteria from the IWSLT evaluation campaign (Akiba et al., 2004), namely word error rate (WER), position-independent word error rate (PER), and the BLEU and NIST scores (Papineni et al., 2002; Doddington, 2002). The two scores measure accuracy, i. e. larger scores are better. The error rates and scores were computed with respect to *m*ultiple reference transla-

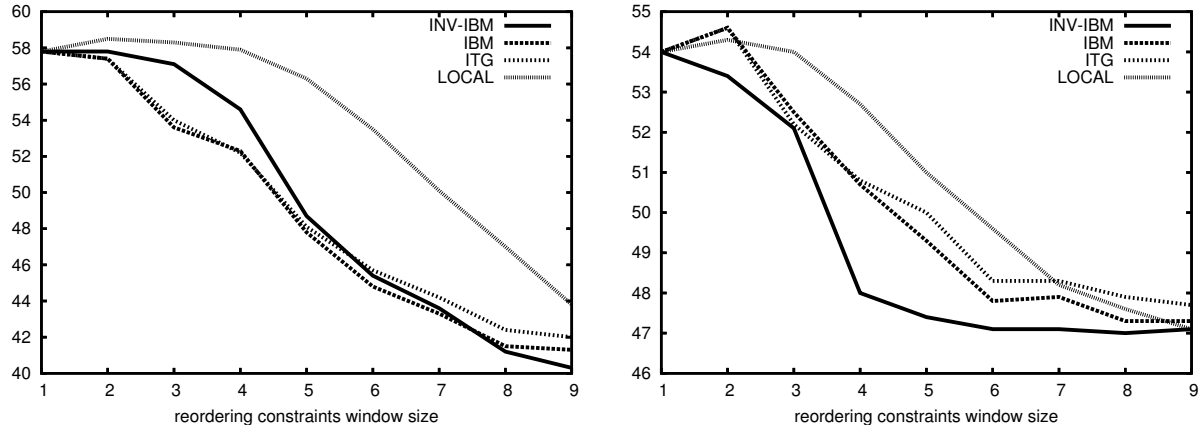


Figure 3: Word error rate [%] as a function of the reordering window size for different reordering constraints: Japanese-to-English (left) and Chinese-to-English (right) translation.

tions, when they were available. To indicate this, we will label the error rate acronyms with an m. Both training and evaluation were performed using corpora and references in lowercase and without punctuation marks.

5.3 Experiments

We used reordering and alignment monotonization in training as described in Sec. 3. To estimate the matrices of local alignment costs for the sentence pairs in the training corpus we used the state occupation probabilities of GIZA++ IBM-4 model training and interpolated the probabilities of source-to-target and target-to-source training directions. After that we estimated a smoothed 4-gram language model on the level of bilingual tuples $f_j, \tilde{e}_j$ and represented it as a finite-state transducer.

When translating, we applied moderate beam pruning to the search automaton only when using reordering constraints with window sizes larger than 3. For very large window sizes we also varied the pruning thresholds depending on the length of the input sentence. Pruning allowed for fast translations and reasonable memory consumption without a significant negative impact on performance.

In our first experiments, we tested the four reordering constraints with various window sizes. We aimed at improving the translation results on the development corpora and compared the results with two baselines: reordering only the source training sentences and translation of the unreordered test sentences; and the GIATI technique for creating bilingual tuples $(f_j, \tilde{e}_j)$ without reordering of the source sentences, neither in training nor during translation.

5.3.1 Highly Non-Monotonic Translation (JE)

Fig. 3 (left) shows word error rate on the Japanese-to-English task as a function of the window size for different reordering constraints. For each of the constraints, good results are achieved using a window size of 9 and larger. This can be attributed to the Japanese word order which is very different from English and often follows a subject-object-verb structure. For small window sizes, ITG or IBM constraints are better suited for this task, for larger window sizes, inverse IBM constraints perform best. The local constraints perform worst and require very large window sizes to capture the main word order differences between Japanese and English. However, their computational complexity is low; for instance, a system with local constraints and window size of 9 is as fast (25 words per second) as the same system with IBM constraints and window size of 5. Using window sizes larger than 10 is computationally expensive and does not significantly improve the translation quality under any of the constraints.

Tab. 2 presents the overall improvements in translation quality when using the best setting: inverse IBM constraints, window size 9. The baseline without reordering in training and testing failed completely for this task, producing empty translations for 37 % of the sentences[2]. Most of the original alignments in training were non-monotonic which resulted in mapping of almost all Japanese words to ε when using only the GIATI monotonization technique. Thus, the proposed reordering methods are of crucial importance for this task.

[2]Hence a NIST score of 0 due to the brevity penalty.

Reordering:	mWER [%]	mPER [%]	BLEU [%]	NIST
BTEC Japanese-to-English (JE) dev				
none	59.7	58.8	13.0	0.00
in training	57.8	39.4	14.7	3.27
+ 9-inv-ibm	40.3	32.1	45.1	8.59
+ rescoring*	39.1	30.9	53.2	9.93
BTEC Chinese-to-English (CE) dev				
none	55.2	52.1	24.9	1.34
in training	54.0	42.3	23.0	4.18
+ 7-inv-ibm	47.1	39.4	34.5	6.53
+ rescoring*	48.3	40.7	39.1	8.11

Table 2: Translation results with optimal reordering constraints and window sizes for the BTEC Japanese-to-English and Chinese-to-English development corpora. *Optimized for the NIST score.

	mWER [%]	mPER [%]	BLEU [%]	NIST
BTEC Japanese-to-English (JE) test				
AT	41.9	33.8	45.3	9.49
WFST	42.1	35.6	47.3	9.50
BTEC Chinese-to-English (CE) test				
AT	45.6	39.0	40.9	8.55
WFST	46.4	38.8	40.8	8.73

Table 3: Comparison of the IWSLT-2004 automatic evaluation results for the described system (WFST) with those of the best submitted system (AT).

Further improvements were obtained with a rescoring procedure. For rescoring, we produced a k-best list of translation hypotheses and used the word penalty and deletion model features, the IBM Model 1 lexicon score, and target language n-gram models of the order up to 9. The scaling factors for all features were optimized on the development corpus for the NIST score, as described in (Bender et al., 2004).

5.3.2 Moderately Non-Mon. Translation (CE)

Word order in Chinese and English is usually similar. However, a few word reorderings over quite large distances may be necessary. This is especially true in case of questions, in which question words like "where" and "when" are placed at the end of a sentence in Chinese. The BTEC corpora contain many sentences with questions.

The inverse IBM constraints are designed to perform this type of reordering (see Sec. 4.3). As shown in Fig. 3, the system performs well under these constraints already with relatively small window sizes. Increasing the window size beyond 4 for these constraints only marginally improves the translation error measures for both short (under 8 words) and long sentences. Thus, a suitable language-pair-specific choice of reordering constraints can avoid the huge computational complexity required for permutations of long sentences.

Reordering:	mWER [%]	mPER [%]	BLEU [%]	NIST
none	25.6	22.0	62.1	10.46
in training	28.0	22.3	58.1	10.32
+ 4-local	26.3	20.3	62.2	10.81
+ weights	25.3	20.3	62.6	10.79
+ 3-ibm	27.2	20.5	61.4	10.76
+ weights	25.2	20.3	62.9	10.80
+ rescoring*	22.2	19.0	69.2	10.47

Table 4: Translation results with optimal reordering constraints and window sizes for the test corpus of the BTEC IE task. *Optimized for WER.

Tab. 2 includes error measures for the best setup with inverse IBM constraints with window size of 7, as well as additional improvements obtained by a k-best list rescoring.

The best settings for reordering constraints and model scaling factors on the development corpora were then used to produce translations of the IWSLT Japanese and Chinese test corpora. These translations were evaluated against multiple references which were unknown to the authors. Our system (denoted with WFST, see Tab. 3) produced results competitive with the results of the best system at the evaluation campaign (denoted with AT (Bender et al., 2004)) and, according to some of the error measures, even outperformed this system.

5.3.3 Almost Monotonic Translation (IE)

The word order in the Italian language does not differ much from the English. Therefore, the absolute translation error rates are quite low and translating without reordering in training and search already results in a relatively good performance. This is reflected in Tab. 4. However, even for this language pair it is possible to improve translation quality by performing reordering both in training and during translation. The best performance on the development corpus is obtained when we constrain the reodering with relatively small window sizes of 3 to 4 and use either IBM or local reordering constraints.

On the test corpus, as shown in Tab. 4, all error measures can be improved with these settings.

Especially for languages with similar word order it is important to use *weighted* reorderings (Sec. 4.6) in order to prefer the original word order. Introduction of reordering weights for this task results in notable improvement of most error measures using either the IBM or local constraints. The optimal probability α for the unreordered path was determined on the development corpus as 0.5 for both of these constraints. The results on the test corpus using this setting are also given in Tab. 4.

6 Conclusion

In this paper, we described a reordering framework which performs source sentence reordering on word level. We suggested to use optimal alignment functions for monotonization and improvement of translation model training. This allowed us to translate monotonically taking a reordering graph as input. We then described known and novel reordering constraints and their efficient finite-state implementations in which the reordering graph is computed on-demand. We also utilized weighted permutations. We showed that our monotonic phrase-based translation approach effectively makes use of the reordering framework to produce quality translations even from languages with significantly different word order. On the Japanese-to-English and Chinese-to-English IWSLT tasks, our system performed at least as well as the best machine translation system.

Acknowledgement

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under the project "Statistische Textübersetzung" (Ne572/5) and by the European Union under the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738).

References

Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii. 2004. *Overview of the IWSLT04 Evaluation Campaign*. Proc. Int. Workshop on Spoken Language Translation, pp. 1–12, Kyoto, Japan.

S. Bangalore and G. Riccardi. 2000. *Stochastic Finite-State Models for Spoken Language Machine Translation*. Proc. Workshop on Embedded Machine Translation Systems, pp. 52–59.

O. Bender, R. Zens, E. Matusov, and H. Ney. 2004. *Alignment Templates: the RWTH SMT System*. Proc. Int. Workshop on Spoken Language Translation, pp. 79–84, Kyoto, Japan.

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. 1996. *Language Translation Apparatus and Method of Using Context-based Translation Models*. United States Patent 5510981.

F. Casacuberta and E. Vidal. 2004. Machine Translation with Inferred Stochastic Finite-State Transducers. Computational Linguistics, vol. 30(2):205-225.

G. Doddington. 2002. *Automatic Evaluation of Machine Translation Quality Using n-gram Co-Occurrence Statistics*. Proc. Human Language Technology Conf., San Diego, CA.

S. Kanthak and H. Ney. 2004. *FSA: an Efficient and Flexible C++ Toolkit for Finite State Automata using On-demand Computation*. Proc. 42nd Annual Meeting of the Association for Computational Linguistics, pp. 510–517, Barcelona, Spain.

K. Knight and Y. Al-Onaizan. 1998. *Translation with Finite-State Devices*. Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1529, pp. 421–437.

S. Kumar and W. Byrne. 2003. *A Weighted Finite State Transducer Implementation of the Alignment Template Model for Statistical Machine Translation*. Proc. Human Language Technology Conf. NAACL, pp. 142–149, Edmonton, Canada.

E. Matusov, R. Zens, and H. Ney. 2004. *Symmetric Word Alignments for Statistical Machine Translation*. Proc. 20th Int. Conf. on Computational Linguistics, pp. 219–225, Geneva, Switzerland.

F. J. Och and H. Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, vol. 29, number 1, pp. 19–51.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proc. 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp. 311–318.

J. M. Vilar, 2000. *Improve the Learning of Subsequential Transducers by Using Alignments and Dictionaries*. Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 1891, pp. 298–312.

D. Wu. 1997. *Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora*. Computational Linguistics, 23(3):377–403.

R. Zens, F. J. Och and H. Ney. 2002. *Phrase-Based Statistical Machine Translation*. In: M. Jarke, J. Koehler, G. Lakemeyer (Eds.): KI - Conference on AI, KI 2002, Vol. LNAI 2479, pp. 18-32, Springer Verlag.

R. Zens and H. Ney. 2003. *A Comparative Study on Reordering Constraints in Statistical Machine Translation*. Proc. Annual Meeting of the Association for Computational Linguistics, pp. 144–151, Sapporo, Japan.

Gaming Fluency: Evaluating the Bounds and Expectations of Segment-based Translation Memory

John Henderson and **William Morgan**
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
{jhndrsn,wmorgan}@mitre.org

Abstract

Translation memories provide assistance to human translators in production settings, and are sometimes used as first-pass machine translation in assimilation settings because they produce highly fluent output very rapidly. In this paper, we describe and evaluate a simple whole-segment translation memory, placing it as a new lower bound in the well-populated space of machine translation systems. The result is a new way to gauge how far machine translation has progressed compared to an easily understood baseline system.

The evaluation also sheds light on the evaluation metric and gives evidence showing that gaming translation with perfect fluency does not fool BLEU the way it fools people.

1 Introduction and background

Translation Memory (TM) systems provide roughly concordanced results from an archive of previously translated materials. They are typically used by translators who want computer assistance for searching large archives for tricky translations, and also to help ensure a group of translators rapidly arrive at similar terminology (Macklovitch et al., 2000). Several companies provide commercial TMs and systems for using and sharing them. TMs can add value to computer assisted translation services (Drugan, 2004).

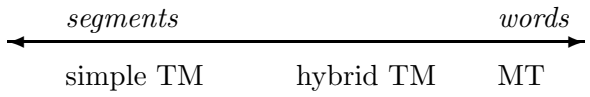
Machine Translation (MT) developers make use of similar historical archives (parallel texts, bitexts), to produce systems that perform a task very similar to TMs. But while TM systems and MT systems can appear strikingly similar, (Marcu, 2001) key differences exist in how they are used.

TMs often need to be fast because they are typically used interactively. They aim to produce highly readable, fluent output, usable in document production settings. In this setting, errors of omission are more easily forgiven than errors of commission so, just like MT, TM output must look good to users who have no access to the information in source texts.

MT, on the other hand, is often used in assimilation settings, where a batch job can often be run on multiple processors. This permits variable rate output and allows slower systems that produce better translations to play a part. Batch MT serving a single user only needs to run at roughly the same rate the reader can consume its output.

Simple TMs operate on an entire translation segment, roughly the size of a sentence or two, while more sophisticated TMs operate on units of varying size: a word, a phrase, or an entire segment (Callison-Burch et al., 2004). Modern approaches to MT, especially statistical MT, typically operate on more fine-grained units, words and phrases (Och and Ney, 2004). The relationship between whole segment TM and MT can be viewed as a continuum of translation granularity:

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 175–182,
Ann Arbor, June 2005.



Simple TM systems, focusing on segment-level granularity, lie at one extreme, and word-for-word, IBM-model MT systems on the other. Example-Based MT (EBMT), phrase-based, and commercial TM systems likely lie somewhere in between.

This classification motivates our work here. MT systems have well-studied and popular evaluation techniques such as BLEU (Papineni et al., 2001). In this paper we lay out a methodology for evaluating TMs along the lines of MT evaluation. This allows us to measure the raw relative value of TM and MT as translation tools, and to develop expectations for how TM performance increases as the size of the memory increases.

There are many ways to perform TM segmentation and phrase extraction. In this study, we use the most obvious and simple condition—a full segment TM. This gives a lower bound on real TM performance, but a lower bound which is not trivial.

Section 2 details the architecture of our simple TM. Section 3 describes experiments involving different strategies for IR, oracle upper bounds on TM performance as the memory grows, and techniques for rescoring the retrievals. Section 4 discusses the results of the experiments.

2 A Simple Chinese-English Translation Memory

For our experiments below, we constructed a simple translation memory from a sentence-aligned parallel corpus. The system consists of three stages. A source-language input string is rewritten to form an information retrieval (IR) query. The IR engine is called to return a list of candidate translation pairs. Finally a single target-language translation as output is chosen.

2.1 Query rewriting

To retrieve a list of translation candidates from the IR engine, we first create a query which is a concatenation of all possible ngrams of the source sentence, for all ngram sizes from 1 to a fixed n.

We rely on the fact that the Chinese data in the translation memory is tokenized and indexed at the unigram level. Each Chinese character in the source sentence is tokenized individually, and we make use of the IR engine's phrase query feature, which matches documents in which all terms in the phrase appear in consecutive order, to create the ngrams. For example, to produce a trigram + bigram + unigram query for a Chinese sentence of 10 characters, we would create a query consisting of eight three-character phrases, nine two-character phrases, and 10 single-character "phrases". All phrases are weighted equally in the query.

This approach allows us to perform lookups for arbitrary ngram sizes. Depending on the specifics of how *idf* is calculated, this may yield different results from indexing ngrams directly, but it is advantageous in terms of space consumed and scalability to different ngram sizes without reindexing.

This is a slight generalization of the successful approach to Chinese information retrieval using bigrams (Kwok, 1997). Unlike that work, we perform no second stage IR after query expansion. Using a segmentation-independent engineering approach to Chinese IR allows us to sidestep the lack of a strong segmentation standard for our heterogeneous parallel corpus and prepares us to rapidly move to other languages with segmentation or lemmatization challenges.

2.2 The IR engine

Simply for performance reasons, an IR engine, or some other sort of index, is needed to implement a TM (Brown, 2004). We use the open-source Lucene $v1.4.3$, (Apa, 2004) as our IR engine. Lucene scores candidate segments from the parallel text using a modified *tf-idf* formula that includes normalizations for the input segment length and the candidate segment length. We did not modify any Lucene defaults for these experiments.

To form our translation memory, we indexed all sentence pairs in the translation memory corpora, each pair as a separate document. We

Source
库林斯说："他随时都可能亮相，这一切取决于他的感觉。 目前，他训练的侧重点是防守，同时也练习投篮。 他准备略为提高强度，以检验自己的身体是否能适应。 就膝盖而言，他说至少比手术前好了百分之百。"
TM output
However , everything depended on the missions to be decided by the Security Council . The presentations focused on the main lessons learned from their activities in the field . It is wrong to commit suicide or to use ones own body as a weapon of destruction . There was practically full employment in all sectors .
One reference translation (of four)
Doug Collins said, "He may appear any time. It really depends on how he feels." At present, his training is defense oriented but he also practices shots. He is elevating the intensity to test whether his body can adapt to it. So far as his knee is concerned, he thinks it heals a hundred percent after the surgery."

Table 1: Typical TM output. Excerpt from a story about athlete Michael Jordan.

indexed in such a way that IR searches can be restricted to just the source language side or just the target language side.

2.3 Rescoring

The IR engine returns a list of candidate translation pairs based on the query string, and the final stage of the TM process is the selection of a single target-language output sentence from that set.

We consider a variety of selection metrics in the experiments below. For each metric, the source-language side of each pair in the candidate list is evaluated against the original source language input string. The target language segment of the pair with the highest score is then output as the translation.

In the case of automated MT evaluation metrics, which are not necessarily symmetric, the source-language input string is treated as the reference and the source-language side of each pair returned by the IR engine as the hypothesis.

All tie-breaking is done via *tf-idf*, i.e. if multiple entries share the same score, the one ranked higher by the search engine will be output.

Table 1 gives a typical example of how the TM performs. Four contiguous source segments are presented, followed by TM output and finally one of the reference translations for those source segments. The only indicator of the translation quality available to monolingual English speakers is the awkwardness of the segments as a group. By design, the TM performs with perfect fluency at the segment level.

3 Experiments

We performed several experiments in the course of optimizing this TM, all using the same set of parallel texts for the TM database and multiple-reference translation corpus for evaluation. The parallel texts for the TM come from several Chinese-English parallel corpora, all available from the Linguistic Data Consortium (LDC). These corpora are described in Table 2. We discarded any sentence pairs that seemed trivially incomplete, corrupt, or otherwise invalid. In the case of LDC2002E18, in which sentences were aligned automatically and confidence scores produced for each alignment, we dropped all pairs with scores above 9, indicating poor alignment. No duplication checks were performed. Our final corpus contained approximately 7 million sentence pairs and contained 3.2 GB of UTF-8 data.

Our evaluation corpus and reference corpus

come from the data used in the NIST 2002 MT competition. (NIST, 2002). The evaluation corpus is 878 segments of Chinese source text. The reference corpus consists of four independent human-generated reference English translations of the evaluation corpus.

All performance measurements were made using a fast reimplementation of NIST's BLEU. BLEU exhibits a high correlation with human judgments of translation quality when measuring on large sections of text (Papineni et al., 2001). Furthermore, using BLEU allowed us to compare our performance to that of other systems that have been tested with the same evaluation data.

3.1 An upper bound on whole-segment translation memory

Our first experiment was to determine an upper bound for the entire translation memory corpus. In other words, given an oracle that picks the best possible translation from the translation memory corpus for each segment in the evaluation corpus, what is the BLEU score for the resulting document? This score is unlikely to approach the maximum, BLEU =100 because this oracle is constrained to selecting a translation from the target language side of the parallel corpus. All of the calculations for this experiment are performed on the target language side of the parallel text.

We were able to take advantage of a trait particular to BLEU for this experiment, avoiding many of BLEU score calculations required to assess all of the 878×7.5 million combinations. BLEU produces a score of 0 for any hypothesis string that doesn't share at least one 4-gram with one reference string. Thus, for each set of four references, we created a Lucene query that returned all translation pairs which matched at least one 4-gram with one of the references. We picked the top segment by calculating BLEU scores against the references, and created a hypothesis document from these segments.

Note that, for document scores, BLEU's brevity penalty (BP) is applied globally to an entire document and not to individual segments. Thus, the document score does not necessarily increase monotonically with increases in scores of individual segments. As more than 99% of the segment pairs we evaluated yielded scores of zero, we felt this would not have a significant effect on our experiments. Also, the TM does not have much liberty to alter the length of the returned segments. Individual segments were chosen to optimize BLEU score, and the resulting documents exhibited appropriately increasing scores. While there is no efficient strategy for whole-document BLEU maximization, an iterative rescoring of the entire document while optimizing the choice of only one candidate segment at a time could potentially yield higher scores than those we report here.

3.2 TM performance with varied Ngram length

The second experiment was to determine the effect that different ngram sizes in the Chinese IR query have on the IR engine's ability to retrieve good English translations.

We considered cumulative ngram sizes from 1 to 7, i.e. unigram, unigram + bigram, unigram + bigram + trigram, and so on. For each set of ngram sizes, we created a Lucene query for every segment of the (Chinese) evaluation corpus. We then produced a hypothesis document by combining the English sides of the top results returned by Lucene for each query. The hypothesis document was evaluated against the reference corpora by calculating a BLEU score.

While it was observed that IR performance is maximized by performing bigram queries (Kwok, 1997), we had reason to believe the TM would not be similar. TMs must attempt to match short sequences of stop words that indicate grammar as well as more traditional content words. Note that our system performed neither stemming nor stop word (or ngram) removal on the input Chinese strings.

3.3 An upper bound on TM N-best list rescoring

The next experiment was to determine an upper bound on the performance of *tf-idf* for different result set sizes, i.e. for different (maximum)

LDC Id	Description	Pairs
LDC2002E18	Xinhua Chinese-English Parallel News Text v. 1.0 beta 2	64,371
LDC2002E58	Sinorama Chinese-English Parallel Text	103,216
LDC2003E25	Hong Kong News Parallel Text	641,308
LDC2004E09	Hong Kong Hansard Parallel Text	1,247,294
LDC2004E12	UN Chinese-English Parallel Text v. 2	4,979,798
LDC2000T47	Hong Kong Laws Parallel Text	302,945
	Total	7,338,932

Table 2: Sentence-aligned parallel corpora used for the creation of the translation memory. The "pairs" column gives the number of translation pairs available after trivial pruning.

numbers of translation pairs returned by the IR engine. This experiment describes the trade-off between more time spent in the IR engine creating a longer list of returns and the potential increase in translation score.

To determine how much IR was "enough" IR, we performed an oracle experiment on different IR query sizes. For each segment of the evaluation corpus, we performed a cumulative 4-gram query as described in Section 4.2. We produced the n-best list oracle's hypothesis document by selecting the English translation from this result set with the highest BLEU score when evaluated against the corresponding segment from the *reference* corpus. We then evaluated the hypothesis documents against the *reference* corpus by computing BLEU scores.

3.4 N-best list rescoring with several MT evaluation metrics

The fourth experiment was to determine whether we could improve upon *tf-idf* by applying automated MT metrics to pick the best sentence from the top n translation pairs returned by the IR engine. We compared a variety of metrics from MT evaluation literatures. All of these were run on the tokens in the source language side of the IR result, comparing against the single pseudo-reference, the original source language segment. While many of these metrics aren't designed to perform well with one reference, they stand in as good approximate string matching algorithms.

The score that the IR engine associates with each segment is retained and marked as *tf-idf* in this experiment. Naturally, BLEU (Papineni et al., 2001) was the first choice metric, as it was well-matched to the target language evaluation function. ROUGE was a reimplementation of ROUGE-L from (Lin and Och, 2004). It computes an F-measure from precision and recall that are both based on the longest common subsequence of the hypothesis and reference strings. WER-G is a variation on traditional word error rate that was found to correlate very well with human judgments (Foster et al., 2003), and PER is the traditional position-independent error rate that was also shown to correlate well with human judgments (Leusch et al., 2003). Finally, a random metric was added to show the BLEU value one could achieve by selecting from the top n strictly by chance.

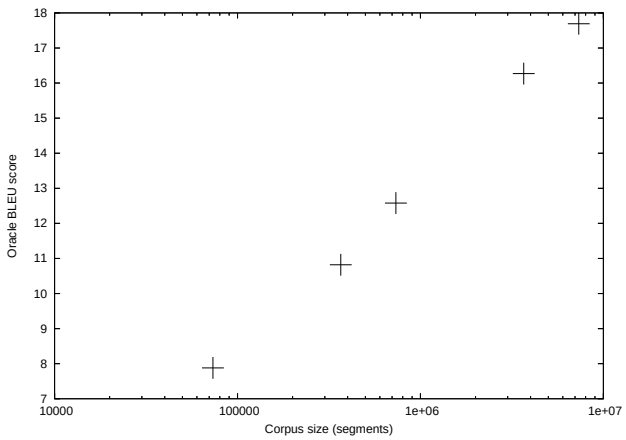
After the individual metrics are calculated for these segments, a uniform-weight log-linear combination of the metrics is calculated and used to produce a new rank ordering under the belief that the different metrics will make predictions that are constructive in aggregate.

4 Results

4.1 An upper bound for whole-sentence TM

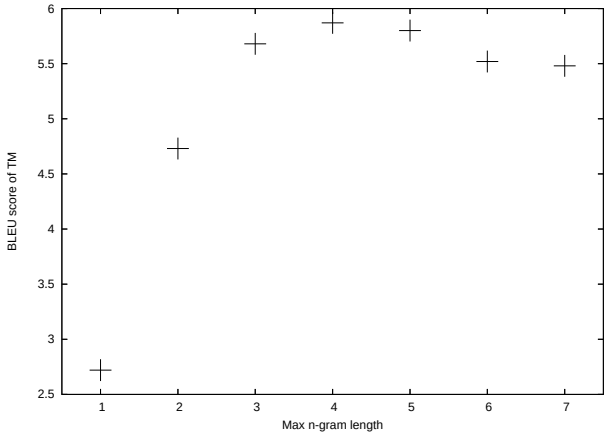
Figure 1 shows the maximum possible BLEU score that can an oracle can achieve by selecting the best English-side segment from the parallel text. The upper bound achieved here is a BLEU score of 17.7, and this number is higher than the best performing system in the corresponding NIST evaluation.

Note the log-linear growth in the resulting



Size	BLEU
73389	7.88
366947	10.82
733893	12.58
3669466	16.27
7338932	**17.69**

Figure 1: Oracle bounds on TM performance as corpus size increases.



Ngrams in query	BLEU
1	2.72
1,2	4.73
1,2,3	5.68
1,2,3,4	**5.87**
1,2,3,4,5	5.80
1,2,3,4,5,6	5.52
1,2,3,4,5,6,7	5.48

Figure 2: BLEU scores for different cumulative ngram sizes, when retrieving only the first translation pair.

BLEU score of the TM with increasing database size. As the database is increased by a factor of ten, the TM gains approximately 5 points of BLEU. While this trend has a natural limit at 20 orders of magnitude, it is unlikely that this amount of text, let alone parallel text, will be a indexed in the foreseeable future. This rate is more useful in interpolation, giving an idea of how much could be gained from adding to corpora that are smaller than 7.5 million segments.

4.2 The effect of ngram size on Chinese *tf-idf* retrieval

Figure 2 shows that our best performance is realized when IR queries are composed of cumulative 4-grams (i.e. unigrams + bigrams + trigrams + 4-grams). As hypothesized, while longer sequences are not important in document retrieval in Chinese IR, they convey information that is useful in segment retrieval in the translation memory. For the remainder of the experiments, we restrict ourselves to cumulative 4-gram queries.

Note that the 4-gram result here (BLEU of 5.87) provides the baseline system performance measure as well as the value when the segments are reranked according to *tf-idf*.

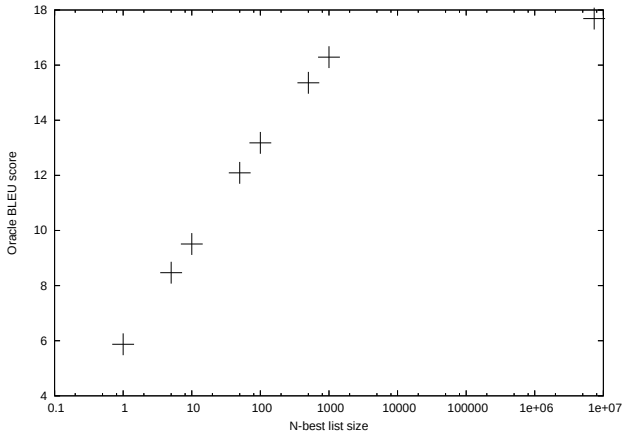
4.3 Upper bounds for *tf-idf*

Figure 3 gives the n-best list rescoring bounds. The upper bound continues to increase up to the top 1000 results. The plateau achieved after 1000 IR results suggests that is little to be gained from further IR engine retrieval.

Note the log-linear growth in the BLEU score the oracle achieves as the n-best list extends on the left side of the figure. As the list length is increased by a factor of ten, the oracle upper bound on performance increases by roughly 3 points of BLEU. Of course, for a system to perform as well as the oracle does becomes progressively harder as the n-best list size increases.

Comparing this result with the experiment in section 4.1 indicates that making the oracle choose among Chinese source language IR results and limiting its view to the 1000 results given by the IR engine incurs only a minor reduction of the oracle's BLEU score, from 17.7 to

16.3. This is one way to measure the impact of crossing this particular language barrier and using IR rather than exhaustive search.



Size	BLEU score
1	5.87
5	8.47
10	9.51
50	12.09
100	13.18
500	15.36
1000	16.29
7338932	**17.69**

Figure 3: BLEU scores for different (maximum) numbers of translation pairs returned by IR engine, where the optimal segment is chosen from the results by an oracle.

4.4 Using automated MT metrics to pick the best TM sentence

Each metric was run on the top 1000 results from the IR engine, on cumulative 4-gram queries. Each metric was given the (Chinese) evaluation corpus segment as the single reference, and scored the Chinese side of each of the 1000 resulting translation pairs against that reference. The hypothesis document for each metric consisted of the English side of the translation pair with the best score for each segment. These documents were scored with BLEU against the reference corpus. Ties (e.g. cases where a metric gave all 1000 pairs the same score) were broken with *tf-idf*.

Results of the rescoring experiment run on an n-best list of size 100 are given in Table 3. Choosing from 1000 pairs did not give better results. Choosing from only 10 gave worse results. The random baseline given in the table represents the expected score from choosing randomly among the top 100 IR returns. While the scores of the individual metrics aside from PER and BLEU reveal no differences, BLEU and the combination metric performed better than the individual metrics.

Metric	BLEU
BLEU	6.20
WER-G	5.90
ROUGE	5.88
tf-idf	5.87
PER	5.72
random	3.32
log(*tf-idf*)	
+log(BLEU)	
+log(ROUGE)	
-log(WER-G)	
-log(PER)	**6.56**

Table 3: BLEU scores for different metrics when picking the best translation from 100 translation pairs returned by the IR engine.

Surprisingly, *tf-idf* was outperformed only by BLEU and the combination metric. While we hoped to gain much more from n-best list rescoring on this task, reaching toward the limits discovered in section 4.3, the combination metric was less than 0.5 BLEU points below the lower range of systems that were entered in the NIST 2002 evals. The BLEU scores of research systems in that competition roughly ranged between 7 and 15. Of course, each of the segments produced by the TM exhibit *perfect fluency*.

5 Discussion

The maximum BLEU score attained by a TM we describe (6.56) would place it in last place in the NIST 2002 evals, but by less than 0.5 BLEU. Successive NIST competitions have exhibited impressive system progress, but each year there have been newcomers who score near (or in some cases lower than) our simple TM baseline.

We have presented several experiments that quantitatively describe how well a simple TM performs when measured with a standard MT evaluation measure, BLEU. We showed that the translation performance of a TM grows as a log-linear function of corpus size below 7.5 million segments. We showed, somewhat surprisingly, only 1000 IR returns need be evaluated by a rescorer to get within 1 BLEU point of the maximum possible score attainable by the TM.

In future work, we expect to validate these results with other language pairs. One question is: how well does this simple IR query expansion address segmented languages and languages that allow more liberal word order? Supervised training of n-best reranking schemes would also determine how far the oracle bound can be pushed. The computationally more expensive reranking procedure that attempts to optimize BLEU on the entire document should be investigated to determine how much can be gained by better global management of the brevity penalty.

Finally, we believe it's worth noting the degree to which high fluency of the TM output could potentially mislead target-language-only readers in their estimation of the system's performance. Table 1 is representative of system output, and is a good example of why translations should not be judged solely on the fluency of a few segments of target language output.

References

Apache Software Foundation, 2004. *Lucene 1.4.3 API*. `http://lucene.apache.org/java/docs/api/`.

Ralf D. Brown. 2004. A modified burrows-wheeler transform for highly-scalable example-based translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation (AMTA-2004)*, Washington, D.C., USA.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2004. Searchable translation memories. In *Proceedings of ASLIB Translation and the Computer 26*.

Joanna Drugan. 2004. Multilingual document management and workflow in the european institutions. In *Proceedings of ASLIB Translation and the Computer 26*.

George Foster, Simona Gandrabur, Cyril Goutte, Erin Fitzgerald, Alberto Sanchis, Nicola Ueffing, John Blatz, and Alex Kulesza. 2003. Confidence estimation for machine translation. Technical report, JHU Center for Language and Speech Processing.

K. L. Kwok. 1997. Comparing representations in chinese information retrieval. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41, New York, NY, USA. ACM Press.

G. Leusch, N. Ueffing, and H. Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proc. of the Ninth MT Summit*, pages 240–247.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, August.

E. Macklovitch, M. Simard, and Ph. Langlais. 2000. Transsearch: A free translation memory on the world wide web. In *Second International Conference On Language Resources and Evaluation (LREC)*, volume 3, pages 1201–1208, Athens Greece, jun.

Daniel Marcu. 2001. Towards a unified approach to memory- and statistical-based machine translation. In *ACL*, pages 378–385.

NIST. 2002. The NIST 2002 machine translation evaluation plan (MT-02). NIST web site. `http://www.nist.gov/speech/tests/mt/doc/2002-MT-EvalPlan-v1.3.pdf`.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

K. Papineni, S. Roukos, T. Ward, and W-J. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Hybrid Example-Based SMT: the Best of Both Worlds?

Declan Groves
School of Computing
Dublin City University
Dublin 9, Ireland
dgroves@computing.dcu.ie

Andy Way
School of Computing
Dublin City University
Dublin 9, Ireland
away@computing.dcu.ie

Abstract

(Way and Gough, 2005) provide an in-depth comparison of their Example-Based Machine Translation (EBMT) system with a Statistical Machine Translation (SMT) system constructed from freely available tools. According to a wide variety of automatic evaluation metrics, they demonstrated that their EBMT system outperformed the SMT system by a factor of two to one.

Nevertheless, they did not test their EBMT system against a phrase-based SMT system. Obtaining their training and test data for English–French, we carry out a number of experiments using the Pharaoh SMT Decoder. While better results are seen when Pharaoh is seeded with Giza++ word- and phrase-based data compared to EBMT sub-sentential alignments, in general better results are obtained when combinations of this 'hybrid' data is used to construct the translation and probability models. While for the most part the EBMT system of (Gough & Way, 2004b) outperforms any flavour of the phrase-based SMT systems constructed in our experiments, combining the data sets automatically induced by both Giza++ and their EBMT system leads to a hybrid system which improves on the EBMT system *per se* for French–English.

1 Introduction

(Way and Gough, 2005) provide what are to our knowledge the first published results comparing Example-Based and Statistical models of Machine Translation (MT). Given that most MT research carried out today is corpus-based, it is somewhat surprising that until quite recently no qualitative research existed on the relative performance of the two approaches. This may be due to a number of factors: the relative unavailability of EBMT systems, the lack of participation of EBMT researchers in competitive evaluations or the dominance in the MT research community of the SMT approach—whenever one paradigm finds favour with the clear majority of MT practitioners, the assumption made by most of the community is that this way of doing things is clearly better than the alternatives.

Like (Way and Gough, 2005), we find this regrettable: the only basis on which such views should be allowed to permeate our field is following extensive testing and evaluation. Nonetheless, given that no EBMT systems are freely available, very few research groups are in the position of being able to carry out such work.

This paper extends the work of (Way and Gough, 2005) by testing EBMT against phrase-based models of SMT, rather than the word-based models used in this previous work. In so doing, it provides a more complete evaluation of the main question at hand, namely whether an SMT system outperforms an EBMT system on reasonably large training and test sets.

We obtained the same training and test data used

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 183–190,
Ann Arbor, June 2005.

in (Way and Gough, 2005), and evaluated a number of SMT systems which use the Pharaoh decoder[1] against the Marker-Based EBMT system of (Gough & Way, 2004b), for French–English and English–French. We provide results using a range of automatic evaluation metrics: BLEU (Papineni et al., 2002), Precision and Recall (Turian et al., 2003), and Word- and Sentence Error Rates. (Way and Gough, 2005) observe that EBMT tends to outperform a word-based SMT model, and our experiments show that a number of different phrase-based SMT systems still tend to fall short of the quality obtained via EBMT for these evaluation metrics. However, when Pharaoh is seeded with the data sets automatically induced by both Giza++ and their EBMT system, better results are seen for French–English than for the EBMT system *per se*.

The remainder of the paper is constructed as follows. In section 2, we summarize the main ideas behind typical models of SMT and EBMT, as well as the EBMT system of (Gough & Way, 2004b) used in our experiments. In section 3, we revisit the experiments and results carried out by (Way and Gough, 2005). In section 4, we describe our extensions to their work, and compare their findings to ours, and in section 5, present a number of hybrid SMT models. Finally, we conclude and offer some thoughts for future work in section 6, and in section 7 present some further comments on the narrowing gap between EBMT and phrase-based SMT.

2 Example-Based and Statistical Models of Translation

A *sine qua non* for both EBMT and SMT is a set of sentences in one language aligned with their translations in another. Although similar in that both models of translation automatically induce translation knowledge from this resource, there are significant differences regarding both the type of information learnt and how this is brought to bear in dealing with new input.

2.1 EBMT

Given a new input string, EBMT models use three separate processes in order to derive translations:

1. Searching the source side of the bitext for 'close' matches and their translations;

2. Determining the sub-sentential translation links in those retrieved examples;

3. Recombining relevant parts of the target translation links to derive the translation.

Searching for the best matches involves determining a similarity metric based on word occurrences and part-of-speech labels, generalised templates and bilingual dictionaries. The recombination process depends on the nature of the examples used in the first place, which may include aligning phrase-structure (sub-)trees (Hearne & Way, 2003) or dependency trees (Watanabe et al., 2003), or using placeables (Brown, 1999) as indicators of chunk boundaries.

Another method—and the one used in the EBMT system used in our experiments—is to use a set of closed-class words to segment aligned source and target sentences and to derive an additional set of lexical and phrasal resources. (Gough & Way, 2004b) base their work on the 'Marker Hypothesis' (Green, 1979), a universal psycholinguistic constraint which posits that languages are 'marked' for syntactic structure at surface level by a closed set of specific lexemes and morphemes. In a preprocessing stage, (Gough & Way, 2004b) use 7 sets of marker words for English and French (e.g. determiners, quantifiers, conjunctions etc.), which together with cognate matches and mutual information scores are used to derive three new data sources: sets of marker chunks, generalised templates and a lexicon.

In order to describe this in more detail, we revisit an example from (Gough & Way, 2004a), namely:

(1) each layer has a layer number $\Longrightarrow$chaque couche a un nombre de la couche

From the sentence pair in (1), the strings in (2) are generated, where marker words are automatically tagged with their marker categories:

[1] http://www.isi.edu/licensed-sw/pharaoh/

(2) <QUANT> each layer has <DET> a layer number $\Longrightarrow$<QUANT> chaque couche a <DET> un nombre <PREP> de la couche

Taking into account marker tag information (label, and relative sentence position), and lexical similarity, the marker chunks in (3) are automatically generated from the marker-tagged strings in (2):

(3) a. <QUANT> each layer has: <QUANT> chaque couche a
b. <DET> a layer number: <DET> un nombre de la couche

(3b) shows that *n:m* alignments are possible (the two French marker chunks *un nombre* and *de la couche* are absorbed into one following the lexical similarities between *layer* and *couche* and *number* and *nombre*, respectively) given the sub-sentential alignment algorithm of (Gough & Way, 2004b).

By generalising over the marker lexicon, a set of marker templates is produced by replacing the marker word by its relevant tag. From the examples in (3), the generalised templates in (4) are derived:

(4) a. <QUANT> layer has: <QUANT> couche a
b. <DET> layer number: <DET> nombre de la couche

These templates increase the robustness of the system and make the matching process more flexible. Now any marker word can be inserted after the relevant tag if it appears with its translation in the lexicon, so that (say) *the layer number* can now be handled by the generalised template in (4b) and inserting a (or all) translation(s) for *the* in the system's lexicon.

2.2 Word- and Phrase-Based SMT

SMT systems require two large probability tables in order to generate translations of new input:

1. a translation model induced from a large amount of bilingual data;

2. a target language model induced from a(n even) large(r) quantity of separate monolingual text.

Essentially, the translation model establishes the set of target language words (and more recently, phrases) which are most likely to be useful in translating the source string, while the language model tries to assemble these words (and phrases) in the most likely target word order. The language model is trained by determining all bigram and/or trigram frequency distributions occurring in the training data, while the translation model takes into account source and target word (and phrase) co-occurrence frequencies, sentence lengths and the relative sentence positions of source and target words.

Until quite recently, SMT models of translation were based on the simple word alignment models of (Brown et al., 1990). Nowadays, however, SMT practitioners also get their systems to learn phrasal as well as lexical alignments (e.g. (Koehn et al., 2003); (Och, 2003)). Unsurprisingly, the quality obtained by today's phrase-based SMT systems is considerably better than that obtained by the poorer word-based models.

3 Comparing EBMT and Word-Based SMT

(Way and Gough, 2005) obtained a large translation memory from *Sun Microsystems* containing 207,468 English–French sentence pairs, of which 3,939 sentence pairs were randomly extracted as a test set, with the remaining 203,529 sentences used as training data. The average sentence length for the English test set was 13.1 words and 15.2 words for the corresponding French test set. The EBMT system used was their Marker-based system as described in section 2.1 above. In order to create the necessary SMT language and translation models, they used:

- *Giza++* (Och & Ney, 2003);[2]
- the CMU-Cambridge statistical toolkit;[3]
- the ISI ReWrite Decoder.[4]

Translation was performed from English–French and French–English, and the resulting translations were evaluated using a range of automatic metrics: BLEU (Papineni et al., 2002), Precision and Recall

[2] http://www.isi.edu/∼och/Giza++.html
[3] http://mi.eng.cam.ac.uk/∼prc14/toolkit.html
[4] http://www.isi.edu/licensed-sw/rewrite-decoder/

(Turian et al., 2003), and Word- and Sentence Error Rates. In order to see whether the amount of training data affected the (relative) performance of the EBMT and SMT systems, (Way and Gough, 2005) split the training data into three sets, of 50K (1.1M words), 100K (2.4M words) and 203K (4.8M words) sentence pairs (TS1–TS3 in what follows).

3.1 English–French Results

Table 1: Comparing the EBMT system of (Gough & Way, 2004b) with a Word-Based SMT (WB-SMT) system for English–French.

		BLEU	Prec.	Recall	WER	SER
TS1	*WB-SMT*	.2971	.6739	.5912	54.9	90.8
	EBMT	.3318	.6525	.6183	54.3	89.2
TS2	*WB-SMT*	.3375	.6824	.5962	51.1	89.9
	EBMT	.4534	.7355	.6983	44.8	77.5
TS3	*WB-SMT*	.3223	.6513	.5704	53.5	89.1
	EBMT	.4409	.6727	.6877	52.4	65.6

The results obtained by (Gough & Way, 2004b) for English–French for their EBMT system and word-based SMT (WB-SMT) are given in Table 1. Essentially, all the automatic evaluation metrics bar one (Precision) suggest that EBMT can outperform SMT from English–French. Surprisingly, however, apart from SER, all evaluation scores are higher using 100K sentence pairs as training data rather than the full 203K sentences. It is generally assumed that increasing the size of the training data for corpus-based MT systems will improve the quality of the output translations. (Way and Gough, 2005) observe that while this dip in performance may be due to a degree of over–fitting, they intend to carry out some variance analysis on these results (e.g. performing bootstrap-resampling on the test set (Koehn, 2004)), or re-test with different sample test sets in order to investigate whether the same phenomenon is observed.

With respect to SER, however, for both SMT and EBMT, the figures improve as more training data is made available. However, the improvement is much more significant for EBMT (20.6%) than for SMT (0.1%). While the WER scores are much the same, indicating that both systems are identifying reasonable target vocabulary that should appear in the output translation, the vast differences in SER using TS3 indicate that a system containing essentially no information about target syntax has very little hope of arranging these target words in the right order. On the contrary, even a system containing some basic knowledge of how phrases fit together such as the Marker-based EBMT system of (Gough & Way, 2004b) will generate translations of far higher quality.

3.2 French–English Results

Table 2: Comparing the EBMT system of (Gough & Way, 2004b) with a WB-SMT system for French–English.

		BLEU	Prec.	Recall	WER	SER
TS1	*WB-SMT*	.3794	.7096	.7355	52.5	86.5
	EBMT	.2571	.5419	.6314	69.7	89.2
TS2	*WB-SMT*	.3924	.7206	.7433	46.2	81.3
	EBMT	.4262	.6731	.7962	55.2	66.2
TS3	*WB-SMT*	.4462	.7035	.7240	46.8	80.8
	EBMT	.4611	.6782	.7441	50.8	51.2

The results obtained by (Way and Gough, 2005) for French–English translations are presented in Table 2. Translating in this language direction is inherently 'easier' than for English–French as far fewer agreement errors and cases of boundary friction are likely. Accordingly, all WB-SMT results in Table 2 are better than for the reverse direction, while for EBMT, improved results are to be seen for BLEU, Recall and SER.

While the majority of metrics obtained for English–French indicate that EBMT outperforms WB-SMT, the results for French–English are by no means as conclusive. Of the 15 tests, WB-SMT outperforms EBMT in nine.

4 Comparing EBMT and Phrase-Based SMT

From the results in the previous sections for French–English and for English–French, (Way and Gough, 2005) observe that EBMT outperforms WB-SMT in the majority of tests. If we are to treat each of the metrics as being equally significant, it can be said that EBMT appears to outperform WB-SMT by a factor of two to one. In fact, the only metric for which EBMT seems to consistently underperform is precision for French–English which, when we examine WER, indicates that the EBMT system's knowledge of word correspondences is incomplete and not as comprehensive as that of the WB-SMT system.

However, it has been apparent for some time now that phrase-based SMT outperforms previous systems using word-based models. The results obtained by (Way and Gough, 2005) for SER also indicate that if phrase-based SMT were used, then improvements in translation quality ought to be seen.

Accordingly, in this section we describe a set of experiments which extends the work of (Way and Gough, 2005) by evaluating the Marker-based EBMT system of (Gough & Way, 2004b) against a phrase-based SMT system built using the following components:

- Giza++, to extract the word-level correspondences;
- The Giza++ word alignments are then refined and used to extract phrasal alignments ((Och & Ney, 2003); or (Koehn et al., 2003) for a more recent implementation);
- Probabilities of the extracted phrases are calculated from relative frequencies;
- The resulting phrase translation table is passed to the Pharaoh phrase-based SMT decoder which along with SRI language modelling toolkit[5] performs translation.

4.1 English–French Results

Table 3: Seeding Pharaoh with Giza++ and EBMT sub-sentential alignments for English–French.

		BLEU	Prec.	Recall	WER	SER
TS3	*GIZA-DATA*	.3753	.6598	.5879	58.5	86.82
	EBMT-DATA	.3643	.6661	.5759	61.33	87.99

We seeded the phrase-based SMT system constructed from the publicly available resources listed above with the word- and phrase-alignments derived via both Giza++ and the Marker-Based EBMT system of (Gough & Way, 2004b). Using the full 203K training set of (Gough & Way, 2004b), and testing on their near 4K test set, the results are given in Table 3. It is clear to see that the Giza++ alignments obtain better scores than the EBMT sub-sentential data. Before one considers the full impact of these results, one should take into account that the size of the EBMT data set (word- and phrase-alignments) is 403,317, while there are over four times as many SMT sub-sentential alignments (1,732,715).

Comparing these results with those in Table 1, we can see that for the same training-test data, the phrase-based SMT system outperforms the WB-SMT system on most metrics, considerably so with respect to BLEU score (.3753 vs. .3223). WER, however, is somewhat worse (.585 vs. .535), and SER remains disappointingly high. Compared to the EBMT system of (Gough & Way, 2004b), the phrase-based SMT system still falls well short with respect to BLEU score (.4409 for EBMT vs. .3573 for SMT), and again, notably for SER (.656 EBMT, .868 SMT).

4.2 French–English Results

Table 4: Seeding Pharaoh with Giza++ and EBMT sub-sentential alignments for French–English.

		BLEU	Prec.	Recall	WER	SER
TS3	*GIZA-DATA*	.4198	.6527	.7100	62.93	82.84
	EBMT-DATA	.3952	.6151	.6643	74.77	86.21

Again, the phrase-based SMT system was seeded with the Giza++ and EBMT alignments, trained on the full 203K training set, and tested on the 4K test set. The results are given in Table 4. As for English–French, the Giza++ alignments obtain better scores than when the EBMT sub-sentential data is used.

Comparing these results with those in Table 2, we see that the phrase-based SMT system actually does worse than WB-SMT, which is an unexpected result[6]. As expected, therefore, the results for phrase-based SMT here are worse still compared to EBMT.

5 Towards Hybridity: Merging SMT and EBMT Alignments

We decided to experiment further by combining parts of the EBMT sub-sentential alignments with parts of the data induced by Giza++. In the following sections, for both English–French and French–English, we seed the Pharaoh phrase-based SMT system with:

[5] http://www.speech.sri.com/projects/srilm/

[6] The Pharaoh system is untuned, so as to provide an easily replicable baseline for other similar research. It is quite possible that with tuning the phrase-based SMT system will outperform the word-based system.

1. the EBMT phrase-alignments with the Giza++ word-alignments;
2. all the EBMT and Giza++ sub-sentential alignments (both words and phrases).

5.1 Giza++ Words and EBMT Phrases

Here we seeded Pharaoh with the word-alignments induced by Giza++ and the EBMT phrasal chunks only (i.e. no Giza++ phrases and no EBMT lexical alignments).

5.1.1 English–French Results

Table 5: Seeding Pharaoh with Giza++ word and EBMT phrasal alignments for English–French.

	BLEU	Prec.	Recall	WER	SER
TS3	.3962	.6773	.5913	59.32	85.43

Using the full 203K training set of (Gough & Way, 2004b), and testing on their near 4K test set, the results are given in Table 5. Comparing these figures to those in Table 3, we can see that all automatic evaluation metrics improve with this hybrid system configuration. Note that the data set size is 430,336, compared to 1.73M for the phrase-based SMT system seeded solely with Giza++ alignments. With respect to the EBMT system *per se* in Table 1, these results remain slightly below those figures (except for precision).

5.1.2 French–English Results

Table 6: Seeding Pharaoh with Giza++ word and EBMT phrasal alignments for French–English.

	BLEU	Prec.	Recall	WER	SER
TS3	.4265	.6424	.6918	68.05	83.40

Running the same experimental set up for the reverse language direction gives the results in Table 6. While recall drops slightly, all the other metrics show a slight increase compared to the performance obtained when Pharaoh is seeded with Giza++ word- and phrase-alignments (cf. Table 4).

5.2 Merging All Data

The following two experiments were carried out by seeding Pharaoh with *all* the EBMT and Giza++ sub-sentential alignments, i.e. both words and phrases.

5.2.1 English–French Results

Table 7: Seeding Pharaoh with all Giza++ and EBMT sub-sentential alignments for English–French.

	BLEU	Prec.	Recall	WER	SER
TS3	.4259	.7026	.6099	54.26	83.63

Inserting all Giza++ and EBMT data into Pharaoh's knowledge sources gives the results in Table 7. These are considerably better than the scores for the 'semi-hybrid' system described in section 5.1.1. This indicates that a phrase-based SMT system is likely to perform better when EBMT word- and phrase-alignments are used in the calculation of the translation and target language probability models. Note, however, that the size of the data set increases to over 2M items. Despite this, compared to the results for the EBMT system of (Gough & Way, 2004b) shown in Table 1, these results for the 'fully hybrid' SMT system still fall somewhat short (except for Precision: .6727 vs. .7026).

5.2.2 French–English Results

Table 8: Seeding Pharaoh with all Giza++ and EBMT sub-sentential alignments for French–English.

	BLEU	Prec.	Recall	WER	SER
TS3	.4888	.6927	.7173	56.37	78.42

Carrying out a similar experiment for the reverse language direction gives the results in Table 8. This time this hybrid SMT system does outperform the EBMT system of (Gough & Way, 2004b), with respect to BLEU score (.4888 vs .4611) and Precision (.6927 vs. 6782), but the EBMT system still wins out where Recall, WER and SER are concerned. Regarding this latter, it seems that the correlation between low SER and high BLEU score is not as important as is claimed in (Way and Gough, 2005).

6 Conclusions

(Way and Gough, 2005) carried out a number of experiments designed to test their large-scale Marker-Based EBMT system described in (Gough & Way, 2004b) against a WB-SMT system constructed from publicly available tools. While the results were a little mixed, the EBMT system won out overall.

Nonetheless, WB-SMT has long been abandoned in favour of phrase-based models. We extended the work of (Way and Gough, 2005) by performing a range of experiments using the Pharaoh phrase-based decoder. Our main observations are as follows:

- Seeding Pharaoh with word- and phrase-alignments induced via Giza++ generates better results than if EBMT sub-sentential data is used.

- Seeding Pharaoh with a 'hybrid' dataset of Giza++ word alignments and EBMT phrases improves over the baseline phrase-based SMT system primed solely with Giza++ data. This would appear to indicate that the quality of the EBMT phrases is better than the SMT phrases, and that SMT practitioners should use EBMT phrasal data in the calculating of their language and translation models, if available.

- Seeding Pharaoh with *all* data induced by Giza++ and the EBMT system leads to the best-performing hybrid SMT system: for English–French, as well as EBMT phrasal data, EBMT word alignments also contribute positively, but the EBMT system *per se* still wins out (except for Precision); for French–English, however, our hybrid Example-Based SMT system outperforms the EBMT system of (Gough & Way, 2004b) (cf. Table 9).

Table 9: Comparing the hybrid phrase-based SMT system using both the full Giza++ and full EBMT data against the EBMT system of (Gough & Way, 2004b) for the full training set (TS3).

		BLEU	**Prec.**	**Recall**	**WER**	**SER**
EN-FR	*HYBRID*	.2971	.6739	.5912	54.9	90.8
	EBMT	.3318	.6525	.6183	54.3	89.2
FR-EN	*HYBRID*	.2971	.6739	.5912	54.9	90.8
	EBMT	.3318	.6525	.6183	54.3	89.2

A number of avenues of further work remain open to us. We would like to extend our investigations into hybrid example-based statistical approaches to machine translation by experiment with seeding the Marker-Based system of (Gough & Way, 2004b) with the SMT data, and combinations thereof with the EBMT sub-sentential alignments, to investigate the effect on translation quality. Given our findings here, we are optimistic that 'hybrid statistical EBMT' will outperform the baseline EBMT system, and that our findings will prompt EBMT practitioners to augment their data resources with SMT alignments, something which to our knowledge is currently not done. In addition, we intend to continue this line of research on different and larger data sets, and for other language pairs.

7 Final Remarks

Finally, as (Way and Gough, 2005) observe, it is difficult to explain why to this day SMT practitioners have not made full use of the large body of existing work on EBMT, from (Nagao, 1984) to (Carl & Way, 2003) and beyond, which has contributed greatly to the field of corpus-based MT.

From its very inception EBMT has made use of a range of sub-sentential data – both phrasal and lexical – to perform translations whereas, until quite recently, SMT models of translation were based on the relatively simple word alignment models of (Brown et al., 1990). With the advent of phrase-based SMT systems the line between EBMT and SMT has become significantly blurred, yet we are still unaware of any papers on SMT which acknowledge their debt to EBMT or which describe their approach as 'example–based'.

Despite it becoming increasingly difficulty to distinguish between EBMT and (phrase–based) SMT models of translation, some differences still exist. Rather than using models of syntax in a *post hoc* fashion, as is the case with most SMT systems, an EBMT model of translation builds in syntax *at its core*. Given this, a phrase–based SMT system is more likely to 'learn' chunks that an EBMT system would not, as the system learns *n*-gram sequences rather than syntactically-motivated phrases *per se*. Furthermore, our research here has demonstrated quite clearly that if available, merging SMT and EBMT data improves the quality of the resulting hybrid SMT system, as phrases extracted by both methods that are more likely to function as syntactic units (and therefore be more beneficial during the translation process) are given a higher statistical significance. Conversely, the probabilities of those 'less useful' SMT *n*-grams that are not also gener-

ated by the EBMT system are reduced. Essentially, the EBMT data helps the SMT system to make the best use of phrase alignments during translation.

Moreover, we see the fact that it is becoming increasingly difficult to describe the differences between EBMT and SMT as a good thing, and that as here, this convergence can lead to hybrid systems capable of outperforming leading EBMT systems as well as state-of-the-art phrase-based SMT.

We hope that the research presented here, together with that begun by (Way and Gough, 2005), will lead to new areas of collaboration between both sets of researchers, to the clear benefit of the MT research community and the wider public.

Acknowledgements

We would like to thank Nano Gough for supplying us with our EBMT training data. Thanks also to three anonymous reviewers for their insightful comments. The work presented in this paper is partly supported by an IRCSET[7] PhD Fellowship Award.

References

Peter Brown, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Fred Jelinek, Robert Mercer, and Paul Roossin. 1990. A statistical approach to machine translation *Computational Linguistics* **16**:79–85.

Ralf Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-based Translation System. In In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, Chester, England, pp.22–32.

Michael Carl and Andy Way (eds). 2003. *Recent Advances in Example-Based Machine Translation*. Kluwer, Dordrecht, The Netherlands.

Nano Gough and Andy Way. 2004. Example-Based Controlled Translation. In *Proceedings of the Ninth EAMT Workshop*, Valetta, Malta, pp.73–81.

Nano Gough and Andy Way. 2004. Robust Large-Scale EBMT with Marker-Based Segmentation. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, Baltimore, MD., pp.95–104.

[7]http://www.ircset.ie

Thomas Green. 1979. The Necessity of Syntax Markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior* **18**:481–496.

Mary Hearne and Andy Way. 2003. Seeing the Wood for the Trees: Data-Oriented Translation. In *MT Summit IX*, New Orleans, LA., pp.165–172.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain, pp.388–395.

Philipp Koehn, Franz Och, and Dan Marcu. 2003. Statistical Phrase-Based Translation. *Human Language Technology Conference, (HLT-NAACL)*, Edmonton, Canada, pp.48–54.

Makoto Nagao. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn and R. Banerji (eds.) *Artificial and Human Intelligence*, North-Holland, Amsterdam, The Netherlands, pp.173–180.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of 41stAnnual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan, pp.160–167.

Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* **29**:19–51.

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA., pp.311–318.

Joseph Turian, Luke Shen and Dan Melamed. 2003. Evaluation of Machine Translation and its Evaluation. In *MT Summit IX*, New Orleans, LA., pp.386–393.

Hideo Watanabe, Sadao Kurohashi and Eiji Aramaki. 2003. Finding Translation Patterns from Paired Source and Target Dependency Structures. In M. Carl & A. Way (eds.) *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp.397–420.

Andy Way and Nano Gough. 2005. Comparing Example-Based and Statistical Machine Translation. *Natural Language Engineering* [in press].

Word Graphs for Statistical Machine Translation

Richard Zens and **Hermann Ney**
Chair of Computer Science VI
RWTH Aachen University
{zens,ney}@cs.rwth-aachen.de

Abstract

Word graphs have various applications in the field of machine translation. Therefore it is important for machine translation systems to produce compact word graphs of high quality. We will describe the generation of word graphs for state of the art phrase-based statistical machine translation. We will use these word graph to provide an analysis of the search process. We will evaluate the quality of the word graphs using the well-known graph word error rate. Additionally, we introduce the two novel graph-to-string criteria: the position-independent graph word error rate and the graph BLEU score.
Experimental results are presented for two Chinese–English tasks: the small IWSLT task and the NIST large data track task. For both tasks, we achieve significant reductions of the graph error rate already with compact word graphs.

1 Introduction

A statistical machine translation system usually produces the single-best translation hypotheses for a source sentence. For some applications, we are also interested in alternative translations. The simplest way to represent these alternatives is a list with the N-best translation candidates. These N-best lists have one major disadvantage: the high redundancy. The translation alternatives may differ only by a single word, but still both are listed completely. Usually, the size of the N-best list is in the range of a few hundred up to a few thousand candidate translations per source sentence. If we want to use larger N-best lists the processing time gets very soon infeasible.

Word graphs are a much more compact representation that avoid these redundancies as much as possible. The number of alternatives in a word graph is usually an order of magnitude larger than in an N-best list. The graph representation avoids the combinatorial explosion that make large N-best lists infeasible.

Word graphs are an important data structure with various applications:

- Word Filter.
  The word graph is used as a compact representation of a large number of sentences. The score information is not contained.
- Rescoring.
  We can use word graphs for rescoring with more sophisticated models, e.g. higher-order language models.
- Discriminative Training.
  The training of the model scaling factors as described in (Och and Ney, 2002) was done on N-best lists. Using word graphs instead could further improve the results. Also, the phrase translation probabilities could be trained discrimatively, rather than only the scaling factors.
- Confidence Measures.
  Word graphs can be used to derive confidence measures, such as the posterior probability (Ueffing and Ney, 2004).

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 191–198,
Ann Arbor, June 2005.

- Interactive Machine Translation.
  Some interactive machine translation systems make use of word graphs, e.g. (Och et al., 2003).

State Of The Art. Although there are these many applications, there are only few publications directly devoted to word graphs. The only publication, we are aware of, is (Ueffing et al., 2002). The shortcomings of (Ueffing et al., 2002) are:

- They use single-word based models only. Current state of the art statistical machine translation systems are phrase-based.
- Their graph pruning method is suboptimal as it considers only partial scores and not full path scores.
- The N-best list extraction does not eliminate duplicates, i.e. different paths that represent the same translation candidate.
- The rest cost estimation is not efficient. It has an exponential worst-case time complexity. We will describe an algorithm with linear worst-case complexity.

Apart from (Ueffing et al., 2002), publications on weighted finite state transducer approaches to machine translation, e.g. (Bangalore and Riccardi, 2001; Kumar and Byrne, 2003), deal with word graphs. But to our knowledge, there are no publications that give a detailed analysis and evaluation of the quality of word graphs for machine translation.

We will fill this gap and give a systematic description and an assessment of the quality of word graphs for phrase-based machine translation. We will show that even for hard tasks with very large vocabulary and long sentences the graph error rate drops significantly.

The remaining part is structured as follows: first we will give a brief description of the translation system in Section 2. In Section 3, we will give a definition of word graphs and describe the generation. We will also present efficient pruning and N-best list extraction techniques. In Section 4, we will describe evaluation criteria for word graphs. We will use the graph word error rate, which is well known from speech recognition. Additionally, we introduce the novel position-independent word graph error rate and the graph BLEU score. These are generalizations of the commonly used string-to-string evaluation criteria in machine translation. We will present experimental results in Section 5 for two Chinese–English tasks: the first one, the IWSLT task, is in the domain of basic travel expression found in phrasebooks. The vocabulary is limited and the sentences are short. The second task is the NIST Chinese–English large data track task. Here, the domain is news and therefore the vocabulary is very large and the sentences are with an average of 30 words quite long.

2 Translation System

In this section, we give a brief description of the translation system. We use a phrase-based translation approach as described in (Zens and Ney, 2004). The posterior probability $Pr(e_1^I|f_1^J)$ is modeled directly using a weighted log-linear combination of a trigram language model and various translation models: a phrase translation model and a word-based lexicon model. These translation models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally, we use a word penalty and a phrase penalty. With the exception of the language model, all models can be considered as within-phrase models as they depend only on a single phrase pair, but not on the context outside of the phrase. The model scaling factors are optimized with respect to some evaluation criterion (Och, 2003).

We extended the monotone search algorithm from (Zens and Ney, 2004) such that reorderings are possible. In our case, we assume that local reorderings are sufficient. Within a certain window, all possible permutations of the source positions are allowed. These permutations are represented as a reordering graph, similar to (Zens et al., 2002). Once we have this reordering graph, we perform a monotone phrase-based translation of this graph. More details of this reordering approach are described in (Kanthak et al., 2005).

3 Word Graphs

3.1 Definition

A word graph is a directed acyclic graph $G = (V, E)$ with one designated root node $n_0 \in V$. The edges are labeled with words and optionally with scores. We will use (n, n', w) to denote an edge from node

n to node n' with word label w. Each path through the word graph represents a translation candidate. If the word graph contains scores, we accumulate the edge scores along a path to get the sentence or string score.

The score information the word graph has to contain depends on the application.

If we want to use the word graph as a word filter, we do not need any score information at all. If we want to extract the single- or N-best hypotheses, we have to retain the string or sentence score information. The information about the hidden variables of the search, e.g. the phrase segmentation, is not needed for this purpose. For discriminative training of the phrase translation probabilities, we need all the information, even about the hidden variables.

3.2 Generation

In this section, we analyze the search process in detail. Later, in Section 5, we will show the (experimental) complexity of each step. We start with the source language sentence that is represented as a linear graph. Then, we introduce reorderings into this graph as described in (Kanthak et al., 2005). The type of reordering should depend on the language pair. In our case, we assume that only local reorderings are required. Within a certain window, all possible reorderings of the source positions are allowed. These permutations are represented as a reordering graph, similar to (Knight and Al-Onaizan, 1998) and (Zens et al., 2002).

Once we have this reordering graph, we perform a monotone phrase-based translation of this graph. This translation process consists of the following steps that will be described afterward:

1. segment into phrase
2. translate the individual phrases
3. split the phrases into words
4. apply the language model

Now, we will describe each step. The first step is the segmentation into phrases. This can be imagined as introducing "short-cuts" into the graph. The phrase segmentation does not affect the number of nodes, because only additional edges are added to the graph.

In the segmented graph, each edge represents a source phrase. Now, we replace each edge with one edge for each possible phrase translation. The edge scores are the combination of the different translation probabilities, namely the within-phrase models mentioned in Section 2. Again, this step does not increase the number of nodes, but only the number of edges.

So far, the edge labels of our graph are phrases. In the final word graph, we want to have words as edge labels. Therefore, we replace each edge representing a multi-word target phrase with a sequence of edges that represent the target word sequence. Obviously, edges representing a single-word phrase do not have to be changed.

As we will show in the results section, the word graphs up to this point are rather compact. The score information in the word graph so far consists of the reordering model scores and the phrase translation model scores. To obtain the sentence posterior probability $p(e_1^I|f_1^J)$, we apply the target language model. To do this, we have to separate paths according to the language model history. This increases the word graph size by an order of magnitude.

Finally, we have generated a word graph with full sentence scores. Note that the word graph may contain a word sequence multiple times with different hidden variables. For instance, two different segmentations into source phrases may result in the same target sentence translation.

The described steps can be implemented using weighted finite state transducer, similar to (Kumar and Byrne, 2003).

3.3 Pruning

To adjust the size of the word graph to the desired density, we can reduce the word graph size using forward-backward pruning, which is well-known in the speech recognition community, e.g. see (Mangu et al., 2000). This pruning method guarantees that the good strings (with respect to the model scores) remain in the word graph, whereas the bad ones are removed. The important point is that we compare the full path scores and not only partial scores as, for instance, in the beam pruning method in (Ueffing et al., 2002).

The forward probabilities $F(n)$ and backward probabilities $B(n)$ of a node n are defined by the

following recursive equations:

$$F(n) = \sum_{(n',n,w)\in E} F(n') \cdot p(n', n, w)$$

$$B(n) = \sum_{(n,n',w)\in E} B(n') \cdot p(n, n', w)$$

The forward probability of the root node and the backward probabilities of the final nodes are initialized with one. Using a topological sorting of the nodes, the forward and backward probabilities can be computed with linear time complexity. The posterior probability $q(n, n', w)$ of an edge is defined as:

$$q(n, n', w) = \frac{F(n) \cdot p(n, n', w) \cdot B(n')}{B(n_0)}$$

The posterior probability of an edge is identical to the sum over the probabilities of all full paths that contain this edge. Note that the backward probability of the root node $B(n_0)$ is identical to the sum over all sentence probabilities in the word graph. Let q^* denoted the maximum posterior probability of all edges and let τ be a pruning threshold, then we prune an edge (n, n', w) if:

$$q(n, n', w) < q^* \cdot \tau$$

3.4 N-Best List Extraction

In this section, we describe the extraction of the N-best translation candidates from a word graph.

(Ueffing et al., 2002) and (Mohri and Riley, 2002) both present an algorithm based on the same idea: use a modified A* algorithm with an optimal rest cost estimation. As rest cost estimation, the negated logarithm of the backward probabilities is used. The algorithm in (Ueffing et al., 2002) has two disadvantages: it does not care about duplicates and the rest cost computation is suboptimal as the described algorithm has an exponential worst-case complexity. As mentioned in the previous section, the backward probabilities can be computed in linear time.

In (Mohri and Riley, 2002) the word graph is represented as a weighted finite state automaton. The word graph is first determinized, i.e. the nondeterministic automaton is transformed in an equivalent deterministic automaton. This process removes the duplicates from the word graph. Out of this determinized word graph, the N best candidates are extracted. In (Mohri and Riley, 2002), ϵ-transitions are ignored, i.e. transitions that do not produce a word. These ϵ-transitions usually occur in the backing-off case of language models. The ϵ-transitions have to be removed *before* using the algorithm of (Mohri and Riley, 2002). In the presence of ϵ-transitions, two path representing the same string are considered equal only if the ϵ-transitions are identical as well.

4 Evaluation Criteria

4.1 String-To-String Criteria

To evaluate the single-best translation hypotheses, we use the following string-to-string criteria: word error rate (WER), position-independent word error rate (PER) and the BLEU score. More details on these standard criteria can be found for instance in (Och, 2003).

4.2 Graph-To-String Criteria

To evaluate the quality of the word graphs, we generalize the string-to-string criteria to work on word graphs. We will use the well-known graph word error rate (GWER), see also (Ueffing et al., 2002). Additionally, we introduce two novel graph-to-string criteria, namely the position-independent graph word error rate (GPER) and the graph BLEU score (GBLEU). The idea of these graph-to-string criteria is to choose a sequence from the word graph and compute the corresponding string-to-string criterion for this specific sequence. The choice of the sequence is such that the criterion is the optimum over all possible sequences in the word graph, i.e. the minimum for GWER/GPER and the maximum for GBLEU.

The **GWER** is a generalization of the word error rate. It is a lower bound for the WER. It can be computed using a dynamic programming algorithm which is quite similar to the usual edit distance computation. Visiting the nodes of the word graph in topological order helps to avoid repeated computations.

The **GPER** is a generalization of the position-independent word error rate. It is a lower bound for the PER. The computation is not as straightforward as for the GWER.

In (Ueffing and Ney, 2004), a method for computing the string-to-string PER is presented. This method cannot be generalized for the graph-to-string computation in a straightforward way. Therefore,

we will first describe an alternative computation for the string-to-string PER and then use this idea for the graph-to-string PER.

Now, we want to compute the number of position-independent errors for two strings. As the word order of the strings does not matter, we represent them as multisets[1] A and B. To do this, it is sufficient to know how many words are in A but not in B, i.e. $a := |A - B|$, and how many words are in B but not in A, i.e. $b := |B - A|$. The number of substitutions, insertions and deletions are then:

$$\begin{aligned} sub &= \min\{a, b\} \\ ins &= a - sub \\ del &= b - sub \\ error &= sub + ins + del \\ &= a + b - \min\{a, b\} \\ &= \max\{a, b\} \end{aligned}$$

It is obvious that there are either no insertions or no deletions. The PER is then computed as the number of errors divided by the length of the reference string.

Now, back to the graph-to-string PER computation. The information we need at each node of the word graph are the following: the remaining multiset of words of the reference string that are not yet produced. We denote this multiset C. The cardinality of this multiset will become the value a in the preceding notation. In addition to this multiset, we also need to count the number of words that we have produced on the way to this node but which are not in the reference string. The identity of these words is not important, we simply have to count them. This count will become the value b in the preceding notation.

If we make a transition to a successor node along an edge labeled w, we remove that word w from the set of remaining reference words C or, if the word w is not in this set, we increase the count of words that are in the hypothesis but not in the reference.

To compute the number of errors on a graph, we use the auxiliary quantity $Q(n, C)$, which is the count of the produced words that are not in the reference. We use the following dynamic programming recursion equations:

$$\begin{aligned} Q(n_0, C_0) &= 0 \\ Q(n, C) &= \min_{n', w:(n', n, w) \in E} \Big\{ Q(n', C \cup \{w\}), \\ &\qquad Q(n', C) + 1 \Big\} \end{aligned}$$

Here, n_0 denote the root node of the word graph, C_0 denotes the multiset representation of the reference string. As already mentioned in Section 3.1, (n', n, w) denotes an edge from node n' to node n with word label w.

In the implementation, we use a bit vector to represent the set C for efficiency reasons. Note that in the worst-case the size of the Q-table is exponential in the length of the reference string. However, in practice we found that in most cases the computation is quite fast.

The **GBLEU** score is a generalization of the BLEU score. It is an upper bound for the BLEU score. The computation is similar to the GPER computation. We traverse the word graph in topological order and store the following information: the counts of the matching n-grams and the length of the hypothesis, i.e. the depth in the word graph. Additionally, we need the multiset of reference n-grams that are not yet produced.

To compute the BLEU score, the n-gram counts are collected over the whole test set. This results in a combinatorial problem for the computation of the GBLEU score. We process the test set sentence-wise and accumulate the n-gram counts. After each sentence, we take a greedy decision and choose the n-gram counts that, if combined with the accumulated n-gram counts, result is the largest BLEU score. This gives a conservative approximation of the true GBLEU score.

4.3 Word Graph Size

To measure the word graph size we use the word graph density, which we define as the number of edges in the graph divided by the source sentence length.

5 Experimental Results

5.1 Tasks

We will show experimental results for two Chinese–English translation tasks.

[1] A multiset is a set that may contain elements multiple times.

Table 1: IWSLT Chinese–English Task: corpus statistics of the bilingual training data.

		Chinese	English
Train	Sentences	20 000	
	Running Words	182 904	160 523
	Vocabulary	7 643	6 982
Test	Sentences	506	
	Running Words	3 515	3 595
	avg. SentLen	6.9	7.1

Table 2: NIST Chinese English task: corpus statistics of the bilingual training data.

		Chinese	English
Train	Sentences	3.2M	
	Running Words	51.4M	55.5M
	Vocabulary	80 010	170 758
Lexicon	Entries	81 968	
Test	Sentences	878	
	Running Words	26 431	23 694
	avg. SentLen	30.1	27.0

IWSLT Chinese–English Task. The first task is the Chinese–English supplied data track task of the International Workshop on Spoken Language Translation (IWSLT 2004) (Akiba et al., 2004). The domain is travel expressions from phrase-books. This is a small task with a clean training and test corpus. The vocabulary is limited and the sentences are relatively short. The corpus statistics are shown in Table 1. The Chinese part of this corpus is already segmented into words.

NIST Chinese–English Task. The second task is the NIST Chinese–English large data track task. For this task, there are many bilingual corpora available. The domain is news, the vocabulary is very large and the sentences have an average length of 30 words. We train our statistical models on various corpora provided by LDC. The Chinese part is segmented using the LDC segmentation tool. After the preprocessing, our training corpus consists of about three million sentences with somewhat more than 50 million running words. The corpus statistics of the preprocessed training corpus are shown in Table 2. We use the NIST 2002 evaluation data as test set.

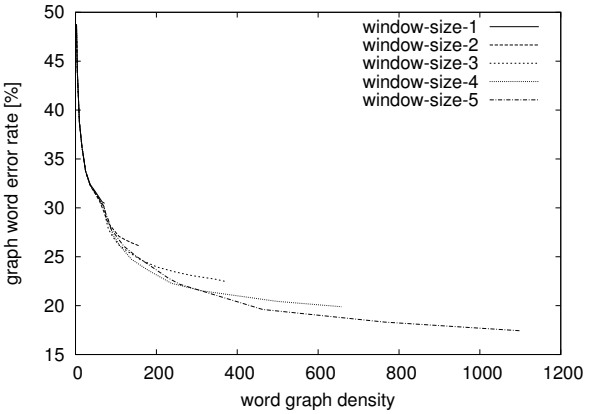


Figure 1: IWSLT Chinese–English: Graph error rate as a function of the word graph density for different window sizes.

5.2 Search Space Analysis

In Table 3, we show the search space statistics of the IWSLT task for different reordering window sizes. Each line shows the resulting graph densities after the corresponding step in our search as described in Section 3.2. Our search process starts with the reordering graph. The segmentation into phrases increases the graph densities by a factor of two. Doing the phrase translation results in an increase of the densities by a factor of twenty. Unsegmenting the phrases, i.e. replacing the phrase edges with a sequence of word edges doubles the graph sizes. Applying the language model results in a significant increase of the word graphs.

Another interesting aspect is that increasing the window size by one roughly doubles the search space.

5.3 Word Graph Error Rates

In Figure 1, we show the graph word error rate for the IWSLT task as a function of the word graph density. This is done for different window sizes for the reordering. We see that the curves start with a single-best word error rate of about 50%. For the monotone search, the graph word error rate goes down to about 31%. Using local reordering during the search, we can further decrease the graph word error rate down to less than 17% for a window size of 5. This is almost one third of the single-best word error rate. If we aim at halving the single-best word error rate, word graphs with a density of less than

Table 3: IWSLT Chinese–English: Word graph densities for different window sizes and different stages of the search process.

language	level	graph type	window size				
			1	2	3	4	5
source	word	reordering	1.0	2.7	6.2	12.8	24.4
	phrase	segmented	2.0	5.0	12.1	26.8	55.6
target		translated	40.8	99.3	229.0	479.9	932.8
	word	TM scores	78.6	184.6	419.2	869.1	1 670.4
		+ LM scores	958.2	2874.2	7649.7	18 029.7	39 030.1

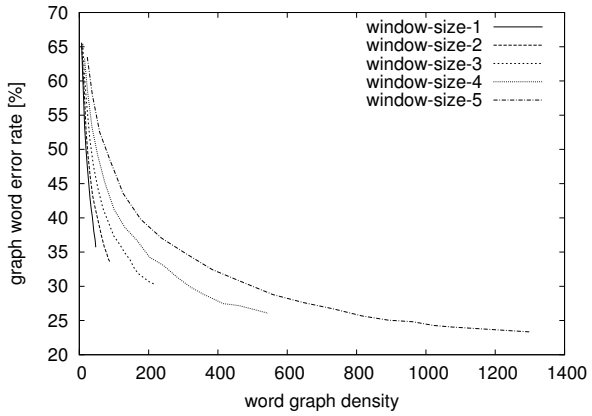


Figure 2: NIST Chinese–English: Graph error rate as a function of the word graph density for different window sizes.

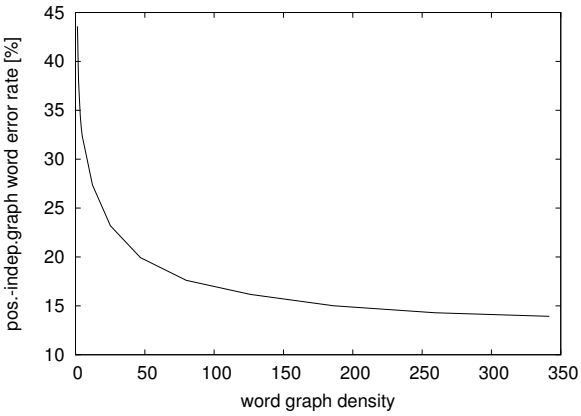


Figure 3: IWSLT Chinese–English: Graph position-independent word error rate as a function of the word graph density.

200 would already be sufficient.

In Figure 2, we show the same curves for the NIST task. Here, the curves start from a single-best word error rate of about 64%. Again, dependent on the amount of reordering the graph word error rate goes down to about 36% for the monotone search and even down to 23% for the search with a window of size 5. Again, the reduction of the graph word error rate compare to the single-best error rate is dramatic. For comparison we produced an N-best list of size $10\,000$. The N-best list error rate (or oracle-best WER) is still 50.8%. A word graph with a density of only 8 has about the same GWER.

In Figure 3, we show the graph position-independent word error rate for the IWSLT task. As this error criterion ignores the word order it is not affected by reordering and we show only one curve. We see that already for small word graph densities the GPER drops significantly from about 42% down to less than 14%.

In Figure 4, we show the graph BLEU scores for the IWSLT task. We observe that, similar to the GPER, the GBLEU score increases significantly already for small word graph densities. We attribute this to the fact that the BLEU score and especially the PER are less affected by errors of the word order than the WER. This also indicates that producing translations with correct word order, i.e. syntactically well-formed sentences, is one of the major problems of current statistical machine translation systems.

6 Conclusion

We have described word graphs for statistical machine translation. The generation of word graphs during the search process has been described in detail. We have shown detailed statistics of the individual steps of the translation process and have given insight in the experimental complexity of each step. We have described an efficient and optimal

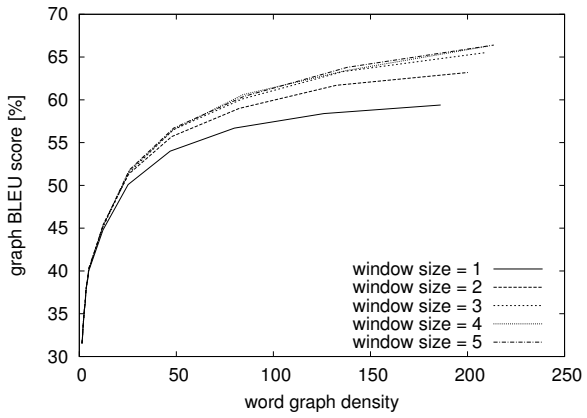


Figure 4: IWSLT Chinese–English: Graph BLEU score as a function of the word graph density.

pruning method for word graphs. Using these technique, we have generated compact word graphs for two Chinese–English tasks. For the IWSLT task, the graph error rate drops from about 50% for the single-best hypotheses to 17% of the word graph. Even for the NIST task, with its very large vocabulary and long sentences, we were able to reduce the graph error rate significantly from about 64% down to 23%.

Acknowledgment

This work was partly funded by the European Union under the integrated project TC-Star (Technology and Corpora for Speech to Speech Translation, IST-2002-FP6-506738, http://www.tc-star.org).

References

Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii. 2004. Overview of the IWSLT04 evaluation campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 1–12, Kyoto, Japan, September/October.

S. Bangalore and G. Riccardi. 2001. A finite-state approach to machine translation. In *Proc. Conf. of the North American Association of Computational Linguistics (NAACL)*, Pittsburgh, May.

S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney. 2005. Novel reordering approaches in phrase-based statistical machine translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, Ann Arbor, MI, June.

K. Knight and Y. Al-Onaizan. 1998. Translation with finite-state devices. In D. Farwell, L. Gerber, and E. H. Hovy, editors, *AMTA*, volume 1529 of *Lecture Notes in Computer Science*, pages 421–437. Springer Verlag.

S. Kumar and W. Byrne. 2003. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pages 63–70, Edmonton, Canada, May/June.

L. Mangu, E. Brill, and A. Stolcke. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer, Speech and Language*, 14(4):373–400, October.

M. Mohri and M. Riley. 2002. An efficient algorithm for the n-best-strings problem. In *Proc. of the 7th Int. Conf. on Spoken Language Processing (ICSLP'02)*, pages 1313–1316, Denver, CO, September.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

F. J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *EACL03: 10th Conf. of the Europ. Chapter of the Association for Computational Linguistics*, pages 387–393, Budapest, Hungary, April.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

N. Ueffing and H. Ney. 2004. Bayes decision rule and confidence measures for statistical machine translation. In *Proc. EsTAL - España for Natural Language Processing*, pages 70–81, Alicante, Spain, October.

N. Ueffing, F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 156–163, Philadelphia, PA, July.

R. Zens and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pages 257–264, Boston, MA, May.

R. Zens, F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *25th German Conf. on Artificial Intelligence (KI2002)*, volume 2479 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 18–32, Aachen, Germany, September. Springer Verlag.

A Recursive Statistical Translation Model*

Juan Miguel Vilar
Dpto. de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Castellón (Spain)
jvilar@lsi.uji.es

Enrique Vidal
Dpto. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Instituto Tecnológico de Informática
Valencia (Spain)
evidal@iti.upv.es

Abstract

A new model for statistical translation is presented. A novel feature of this model is that the alignments it produces are hierarchically arranged. The generative process begins by splitting the input sentence in two parts. Each of the parts is translated by a recursive application of the model and the resulting translation are then concatenated. If the sentence is small enough, a simpler model (in our case IBM's model 1) is applied.

The training of the model is explained. Finally, the model is evaluated using the corpora from a large vocabulary shared task.

1 Introduction

Suppose you were to find an English translation for a Spanish sentence. One possible approach is to assume that every English sentence is a candidate but that different English sentences have different probabilities of being the correct translation. Then, the translation task can be divided in two parts: define an adequate probability distribution that answers to the question "given this English sentence, which is the probability that it is a good translation of that Spanish sentence?"; and use that distribution in order to find the most likely translation of your input sentence.

*Work partially supported by Bancaixa through the project "Sistemas Inductivos, Estadísticos y Estructurales, para la Traducción Automática (Siesta)".

This approach is referred to as the statistical approach to machine translation. The usual approach is to define an statistical model and train its parameters from a training corpus consisting in pairs of sentences that are known to be translation of each other. Different models have been presented in the literature, see for instance (Brown et al., 1993; Och and Ney, 2004; Vidal et al., 1993; Vogel et al., 1996). Most of them rely on the concept of alignment: a mapping from words or groups of words in a sentence into words or groups in the other (in the case of (Vidal et al., 1993) the mapping goes from rules in a grammar for a language into rules of a grammar for the other language). This concept of alignment has been also used for tasks like authomatic vocabulary derivation and corpus alignment (Dagan et al., 1993).

A new statistical model is proposed in this paper, which was initially introduced in (Vilar Torres, 1998). This model is designed so that the alignment between two sentences can be seen in an structured manner: each sentence is divided in two parts and they are put in correspondence; then each of those parts is similarly divided and related to its translation. This way, the alignment can be seen as a tree structure which aligns progressively smaller segments of the sentences. This recursive procedure gives its name to the model: MAR, which comes from "Modelo de Alineamiento Recursivo", which is Spanish for "Recursive Alignment Model".

The rest of the paper is structured as follows: after a comment on previous works, we introduce the notation that we will use throughout the paper, then we briefly explain the model 1 from IBM, next we

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 199–207,
Ann Arbor, June 2005.

introduce our model, then we explain the process of parameter estimation, and how to use the model to translate new test sentences. Finally, we present some experiments and results, together with conclusions.

2 Previous works

The initial formulation of the proposed model, including the training procedures, was presented in (Vilar Torres, 1998), along with preliminary experiments in a small translation task which provided encouraging results.

This model shares some similarities with the stochastic inversion transduction grammars (SITG) presented by Wu in (Wu, 1997). The main point in common is the type of possible alignments considered in both models. Some of the properties of these alignments are studied in (Zens and Ney, 2003). However, the parametrizations of SITGs and the MAR are completely different. The generative process of SITGs produces simultaneously the input and output sentences and the parameters of the model refer to the rules of the nonterminals. This provides a symmetry to both input and output sentences. In contrast, our model clearly distinguishes the input and output sentences and the parameters are based on observable properties of the strings (their lengths and the words composing them). On the other hand, the MAR idea of splitting the sentences until a simple structure is found, also appears in the Divisive Clustering approach presented in (Deng et al., 2004). Again, the main difference lies in the probabilistic modeling of the alignments. In Divisive Clustering a uniform distribution on the alignments is assumed while MAR uses a explicit parametrization.

3 Some notation

In the rest of the paper, we use the following notation. Sentences are taken as concatenations of symbols (words) and represented using a letter and a small bar, like in $\bar{x}$. The individual words are designed by the name of the sentence and a subindex indicating the position, so $\bar{x} = x_1 x_2 \dots x_n$. The length of a sentence is indicated by $|\bar{x}|$. Segments of a sentence are denoted by $\bar{x}_i^j = x_i \dots x_j$. For the substrings of the form $\bar{x}_i^{|\bar{x}|}$ we use the notation $\bar{x}_i$;.

Consistently, $\bar{x}$ denotes the input sentence and $\bar{y}$ its translation and both are assumed to have at least one word. The input and output vocabularies are $\mathcal{X}$ and $\mathcal{Y}$, respectively. Finally, we assume that we are presentend a set $\mathcal{M}$ for training our models. The elements of this set are pairs $(\bar{x}, \bar{y})$ where $\bar{y}$ is a possible translation for $\bar{x}$.

4 IBM's model 1

IBM's model 1 is the simplest of a hierarchy of five statistical models introduced in (Brown et al., 1993). Each model of the hierarchy can be seen as a refinement of the previous ones. Although model 1, which we study here, relies on the concept of alignment, its formulation allows an interpretation of it as a relationship between multisets of words (the order of the words is irrelevant in the final formula).

A word of warning is in order here. The model we are going to present has an important difference with the original: we do not use the empty word. This is a virtual word which does not belong to the vocabulary of the task and that is added to the beginning of each sentence in order to allow words in the output that cannot be justified by the words in the input. We have decided not to incorporate it because of the use we are going to make of the model. As we will see, model 1 is going to be used repeatedly over different substrings of the input sentence in order to analyze their contribution to the total translation. This means that we would have an empty word in each of these substrings. We have decided to avoid this "proliferation" of empty words. Future work may introduce the concept in a more appropriate way.

The model 1 makes two assumptions. That a *stochastic dictionary* can be employed to model the probability that word y is the translation of word x and that all the words in the input sentence have the same weight in producing a word in the output. This leads to:

$$p_I(\bar{y} \mid \bar{x}) = \frac{\varepsilon(|\bar{x}|, |\bar{y}|)}{|\bar{x}|^{|\bar{y}|}} \prod_{j=1}^{|\bar{y}|} \sum_{i=1}^{|\bar{x}|} t(y_j \mid x_i). \qquad (1)$$

Where t is the stochastic dictionary and ε represents a table that relates the length of the alignment with the length of the input sentence (we assume that there is a finite range of possible lengths). This explicit relations between the lengths is not present in

the original formulation of the model, but we prefer to include it so that the probabilities are adequately normalized.

Clearly, this model is not adequate to describe complex translations in which complicated patterns and word order changes may appear. Nevertheless, this model can do a good job to describe the translation of short segments of texts. For example, it can be adequate to model the translation of the Spanish "gracias" into the English "thank you".

5 A Recursive Alignment Model

To overcome that limitation of the model we will take the following approach: if the sentence is complex enough, it will be divided in two and the two halves will be translated independently and joined later; if the sentence is simple, the model 1 will be used.

Let us formalize this intuition for the generative model. We are given an input sentence $\bar{x}$ and the first decission is whether $\bar{x}$ is going to be translated by IBM's model 1 or it is complex enough to be translated by MAR. In the second case, three steps are taken: a cut point of $\bar{x}$ is defined, each of the resulting parts are translated, and the corresponding translations are concatenated. For the translation of the second step, the same process is *recursively* applied. The concatenation of the third step can be done in a "direct" way (the translation of the first part and then the translation of the second) or in an "inverse" way (the translation of the second part and then the translation of the first). The aim of this choice is to allow for the differences in word order between the input and ouput languages.

So, we are proposing an alignment model in which IBM's model 1 will account for translation of elementary segments or individual words while translation of larger and more complex segments or whole sentences will rely on a hierarchical alignment pattern in which model 1 alignments will be on the lowest level of the hierarchy.

Following this discussion, the model can be formally described through a series of four random experiments:

- The first is the selection of the model. It has two possible outcomes: IBM and MAR, with obvious meanings.
- The second is the choice of b, a cut point of $\bar{x}$. The segment $\bar{x}_1^b$ will be used to generate one of the parts of the translation, the segment $\bar{x}_{b+1}$ will generate the other. It takes values from 1 to $|\bar{x}| - 1$.
- The third is the decision about the order of the concatenation. It has two possible outcomes: D (for direct) and I (for inverse).
- The fourth is the translation of each of the halves of $\bar{x}$. They take values in $\mathcal{Y}^+$.

The translation probability can be approximated as follows:

$$p_T(\bar{y} \mid \bar{x}) = \Pr(M = \text{IBM} \mid \bar{x}) p_I(\bar{y} \mid \bar{x}) + \Pr(M = \text{MAR} \mid \bar{x}) p_M(\bar{y} \mid \bar{x}).$$

The value of $p_I(\bar{y} \mid \bar{x})$ corresponds to IBM's model 1 (Equation 1). To derive $p_M(\bar{y} \mid \bar{x})$, we observe that:

$$p_M(\bar{y} \mid \bar{x}) = \sum_{b=1}^{|\bar{x}|-1} \Pr(b \mid \bar{x}) \sum_{d \in \{D,I\}} \Pr(d \mid b, \bar{x}) \sum_{\bar{y}_1 \in \mathcal{Y}^+} \Pr(\bar{y}_1 \mid b, d, \bar{x}) \sum_{\bar{y}_2 \in \mathcal{Y}^+} \Pr(\bar{y}_2 \mid b, d, \bar{x}, \bar{y}_1) \Pr(\bar{y} \mid d, b, \bar{x}, \bar{y}_1, \bar{y}_2).$$

Note that the probability that $\bar{y}$ is generated from a pair $(\bar{y}_1, \bar{y}_2)$ is 0 if $\bar{y} \neq \bar{y}_1 \bar{y}_2$ and 1 if $\bar{y} = \bar{y}_1 \bar{y}_2$, so the last two lines can be rewritten as:

$$\begin{aligned} &\sum_{\bar{y}_1 \in \mathcal{Y}^+} \Pr(\bar{y}_1 \mid b, d, \bar{x}) \sum_{\bar{y}_2 \in \mathcal{Y}^+} \Pr(\bar{y}_2 \mid b, d, \bar{x}, \bar{y}_1) \Pr(\bar{y} \mid b, d, \bar{x}, \bar{y}_1, \bar{y}_2) \\ &= \sum_{\substack{\bar{y}_1, \bar{y}_2 \in \mathcal{Y}^+ \\ \bar{y} = \bar{y}_1 \bar{y}_2}} \Pr(\bar{y}_1 \mid b, d, \bar{x}) \Pr(\bar{y}_2 \mid b, d, \bar{x}, \bar{y}_1) \\ &= \sum_{\bar{y}_1 \,\in\, \text{pref}(\bar{y}) - \bar{y}} \Pr(\bar{y}_1 \mid b, d, \bar{x}) \Pr(\bar{y}_1^{-1} \bar{y} \mid b, d, \bar{x}, \bar{y}_1) \\ &= \sum_{c=1}^{|\bar{y}|-1} \Pr(\bar{y}_1^c \mid b, d, \bar{x}) \Pr(\bar{y}_{c+1} \mid b, d, \bar{x}, \bar{y}_1^c), \end{aligned}$$

where $\mathrm{pref}(\bar{y})$ is the set of *prefixes* of $\bar{y}$. And finally:

$$p_M(\bar{y} \mid \bar{x}) = \sum_{b=1}^{|\bar{x}|-1} \Pr(b \mid \bar{x}) \sum_{d \in \{D,I\}} \Pr(d \mid b, \bar{x}) \sum_{c=1}^{|\bar{y}|-1} \Pr(\bar{y}_1^c \mid b, d, \bar{x}) \Pr(\bar{y}_{c+1}^{\cdot} \mid b, d, \bar{x}, \bar{y}_1^c). \quad (2)$$

The number of parameters of this model is very large, so it is necessary to introduce some simplifications in it. The first one relates to the decision of the *translation model*: we assume that it can be done just on the basis of the length of the input sentence. That is, we cat set up two tables, $\mathcal{M}_I$ and $\mathcal{M}_M$, so that

$$\Pr(M = \text{IBM} \mid \bar{x}) \approx \mathcal{M}_I(|\bar{x}|),$$
$$\Pr(M = \text{MAR} \mid \bar{x}) \approx \mathcal{M}_M(|\bar{x}|).$$

Obviously, for any $\bar{x} \in \mathcal{X}^+$, we will have $\mathcal{M}_I(|\bar{x}|) + \mathcal{M}_M(|\bar{x}|) = 1$. On the other hand, since it is not possible to break a one word sentence, we define $\mathcal{M}_I(1) = 1$. This restriction comes in the line mentioned before: the translation of longer sentences will be structured whereas shorter ones can be translated directly.

In order to decide the *cut point*, we will assume that the probability of cutting the input sentence at a given position b is most influenced by the words around it: x_b and x_{b+1}. We use a table $\mathcal{B}$ such that:

$$\Pr(b \mid \bar{x}) \approx \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})}.$$

This can be interpreted as having a weight for each pair of words and normalizing these weights in each sentence in order to obtaing a proper probability distribution.

Two more tables, $\mathcal{D}_D$ and $\mathcal{D}_I$, are used to store the probabilities that the *alignment be direct or inverse*. As before, we assume that the decission can be made on the basis of the symbols around the cut point:

$$\Pr(d = D \mid b, \bar{x}) = \mathcal{D}_D(x_b, x_{b+1}),$$
$$\Pr(d = I \mid b, \bar{x}) = \mathcal{D}_I(x_b, x_{b+1}).$$

Again, we have $\mathcal{D}_D(x_b, x_{b+1}) + \mathcal{D}_I(x_b, x_{b+1}) = 1$ for every pair of words (x_b, x_{b+1}).

Finally, a probability must be assigned to the translation of the two halves. Assuming that they are independent we can apply the model in a recursive manner:

$$\Pr(\bar{y}_1^c \mid b, d, \bar{x}) \approx \begin{cases} p_T(\bar{y}_1^c \mid \bar{x}_1^b) & \text{if } d = D, \\ p_T(\bar{y}_1^c \mid \bar{x}_{b+1}^{\cdot}) & \text{if } d = I, \end{cases}$$

$$\Pr(\bar{y}_{c+1}^{\cdot} \mid b, d, \bar{x}, \bar{y}_1^c) \approx \begin{cases} p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_{b+1}^{\cdot}) & \text{if } d = D, \\ p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_1^b) & \text{if } d = I. \end{cases}$$

Finally, we can rewrite (2) as:

$$p_M(\bar{y} \mid \bar{x}) = \sum_{b=1}^{|\bar{x}|-1} \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})} \cdot \left(\mathcal{D}_D(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_1^c \mid \bar{x}_1^b) p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_{b+1}^{\cdot}) + \mathcal{D}_I(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_1^b) p_T(\bar{y}_1^c \mid \bar{x}_{b+1}^{\cdot}) \right).$$

The final form of the complete model is then:

$$p_T(\bar{y} \mid \bar{x}) = \mathcal{M}_I(|\bar{x}|) p_I(\bar{y} \mid \bar{x}) + \mathcal{M}_M(|\bar{x}|) \sum_{b=1}^{|\bar{x}|-1} \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})} \cdot \left(\mathcal{D}_D(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_1^c \mid \bar{x}_1^b) p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_{b+1}^{\cdot}) + \mathcal{D}_I(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_1^b) p_T(\bar{y}_1^c \mid \bar{x}_{b+1}^{\cdot}) \right). \quad (3)$$

6 Parameter estimation

Once the model is defined, it is necessary to find a way of estimating its parameters given a training corpus $\mathcal{M}$. We will use maximun likelihood estimation. In our case, the likelihood of the sample corpus is:

$$V = \prod_{(\bar{x}, \bar{y}) \in \mathcal{M}} p_T(\bar{y} \mid \bar{x}).$$

In order to maximize V, initial values are given to the parameters and they are reestimated using repeatedly Baum-Eagon's (Baum and Eagon, 1967) and Gopalakrishnan's (Gopalakrishnan et al., 1991) inequalities. Let P be a parameter of the model (except for those in $\mathcal{B}$) and let $\mathcal{F}(P)$ be its "family" (i.e. the set of parameters such that $\sum_{Q \in \mathcal{F}(P)} Q = 1$). Then, a new value of P can be computed as follows:

$$\begin{aligned}\mathcal{N}(P) &= \frac{P \frac{\partial V}{\partial P}}{\sum_{Q \in \mathcal{F}(P)} Q \frac{\partial V}{\partial Q}} \\ &= \frac{\sum_{(\bar{x},\bar{y}) \in \mathcal{M}} \frac{P}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial P}}{\sum_{Q \in \mathcal{F}(P)} \sum_{(\bar{x},\bar{y}) \in \mathcal{M}} \frac{Q}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial Q}} \\ &= \frac{\mathcal{C}(P)}{\sum_{Q \in \mathcal{F}(P)} \mathcal{C}(Q)},\end{aligned} \tag{4}$$

where

$$\mathcal{C}(P) = \sum_{(\bar{x},\bar{y}) \in \mathcal{M}} \frac{P}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial P}, \tag{5}$$

are the "counts" of parameter P. This is correct as long as V is a polynomial in P. However, we have a problem for $\mathcal{B}$ since V is a rational function of these parameters. We can solve it by assuming, without lose of generality, that $\sum_{x_1, x_2 \in \mathcal{X}} \mathcal{B}(x_1, x_2) = 1$. Then Gopalakrishnan's inequality can be applied similarly and we get:

$$\mathcal{N}(P) = \frac{C + \mathcal{C}(P)}{\sum_{Q \in \mathcal{F}(P)} C + \mathcal{C}(Q)}, \tag{6}$$

where C is an adequate constant. Now it is easy to design a reestimation algorithm. The algorithm gives arbitrary initial values to the parameters (typically those corresponding to uniform probabilities), computes the counts of the parameters for the corpus and, using either (4) or (6), gets new values for the parameters. This cycle is repeated until a stopping criterion (in our case a prefixed number of iterations) is met. This algorithm can be seen in Figure 1

7 Some notes on efficiency

Estimating the parameters as discussed above entails high computational costs: computing $p_T(\bar{y} \mid \bar{x})$ requires $\mathcal{O}(mn)$ arithmetic operations involving the values of $p_T(\bar{y}_i^j \mid \bar{x}_k^l)$ for every possible value of i, j, k and l, which are $\mathcal{O}(m^2 n^2)$. This results in a global cost of $\mathcal{O}(m^3 n^3)$. On the other hand, computing $\frac{\partial p_T}{\partial P}$ costs as much as computing p_T. So it is interesting to keep the number of computed derivatives low.

7.1 Reduction of the parameters to train

In the experiments we have followed some heuristics in order not to reestimate certain parameters:

- The values of M_I —and, consequently, of M_M— for lengths higher than a threshold are assumed to be 0 and therefore there is no need to estimate them.
- As a consequence, the values of ε for lengths above the same threshold, need not be reestimated.
- The values of t for pairs of words with counts under a certain threshold are not reestimated.

Furthermore, during the computation of counts, the recursion is cut on those substring pairs where the value of the probability for the translation is very small.

7.2 Efficient computation of model 1

Other source of optimization is the realization that for computing $p_T(\bar{y} \mid \bar{x})$, it is necessary to compute the value of p_I for each possible pair $(\bar{x}_{ib}^{ie}, \bar{y}_{ob}^{oe})$ (where ib, ie, ob and oe stand for *input begin*, *input end*, *output begin* and *output end*, respectively). Fortunately, it is possible to accelerate this computations. First, define:

$$\begin{aligned}I(ib, ie, ob, oe) &= \frac{p_I(\bar{x}_{ib}^{ie}, \bar{y}_{ob}^{oe})}{\varepsilon(ie - ib + 1, oe - ob + 1)} \\ &= \frac{1}{(ie - ib + 1)^{oe - ob + 1}} \prod_{j=ob}^{oe} \sum_{i=ib}^{ie} t(\bar{y}_j \mid \bar{x}_i).\end{aligned}$$

Now let

$$S(ib, ie, j) = \sum_{i=ib}^{ie} t(\bar{y}_j \mid \bar{x}_i).$$

Algorithm *Maximum likelihood estimation*
give initial values to the parameters;
repeat
initialize the counts to 0;
for each $(\bar{x}, \bar{y}) \in \mathcal{M}$ **do**
compute $p_T(\bar{y} \mid \bar{x})$;
for each parameter P involved in the alignment of $(\bar{x}, \bar{y})$ **do**

$$\mathcal{C}_P := \mathcal{C}_P + \frac{P}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial\, P};$$

endfor
endfor
for each parameter P **do**
reestimate P using (4) or (6);
endfor
until the stopping criterion is met;
End *Maximum likelihood estimation*

Figure 1: Algorithm for maximum likelihood estimation of the parameters of MAR

This leads to

$$I(ib, ie, ob, oe) = S(ib, ie, ob),$$

if $ob = oe$, and to

$$I(ib, ie, ob, oe) = \frac{I(ib, ie, ob, oe-1) S(ib, ie, ob)}{(ie - ib + 1)},$$

if $ob \neq oe$.

So we can compute all values of I with the algorithm in Figure 2.

7.3 Splitting the corpora

Another way of reducing the costs of training has been the use of a heuristic to split long sentences into smaller parts with a length less than l words.

Suppose we are to split sentences $\bar{x}$ and $\bar{y}$. We begin by aligning each word in $\bar{y}$ to a word in $\bar{x}$. Then, a score and a translation is assigned to each substring $\bar{x}_i^j$ with a length below l. The translation is produced by looking for the substring of $\bar{y}$ which has a length below l and which has the largest number of words aligned to positions between i and j. The pair so obtained is given a score equal to sum of: (a) the square of the length of $\bar{x}_i^j$; (b) the square of the number of words in the output aligned to the input; and (c) minus ten times the sum of the square of the number of words aligned to a nonempty position out of $\bar{x}_i^j$ and the number of words outside the segment chosen that are aligned to $\bar{x}_i^j$.

After the segments of $\bar{x}$ are so scored, the partition of $\bar{x}$ that maximizes the sum of scores is computed by dynamic programming.

8 Translating the test sentences

The MAR model can be used to obtain adequate bilingual templates which can be used to translate new test sentences using an appropriate template-based translation system. Here we have adopted the `pharaoh` program (Koehn, 2004).

8.1 Finding the templates

The parameters of the MAR were trained using the algorithm above: first ten IBM model 1 iterations were used for giving initial values to the dictionary probabilities and then five more iterations for re-training the dictionary together with the rest of the parameters.

The alignment of a pair has the form of a tree similar to the one in Figure 3 (this is one of the sentences from the Spanish-English part of the training corpus). Each interior node has two children corresponding to the translation of the two parts in which the input sentence is divided. The leaves of the tree correspond to those segments that were translated by model 1. The templates generated were those defined by the leaves. Further templates were obtained by interpreting each pair of words in the dictionary as a template.

Algorithm all IBM
  for $ob := 1$ **to** $|\bar{y}|$ **do**
    for $oe := ob$ **to** $|\bar{y}|$ **do**
      for $ib := 1$ **to** $|\bar{x}|$ **do**
        $S := 0$;
        for $ie := ib$ **to** $|\bar{x}|$ **do**
          $S := S + t(y_{oe} \mid x_{ie})$;
          $$I(ib, ie, ob, oe) := \begin{cases} S/(ie - ib + 1) & \text{if } ob = oe, \\ I(ib, ie, ob, oe - 1) \times S/(ie - ib + 1) & \text{otherwise;} \end{cases}$$
End all IBM

Figure 2: Efficient computation of different values of IBM's model 1.

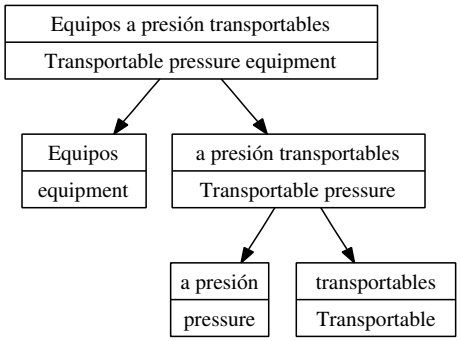


Figure 3: A sample alignment represented as a tree.

Each template was assigned four weights[1] in order to use the `pharaoh` program. For the templates obtained from the alignments, the first weight was the probability assigned to it by MAR, the second weight was the count for the template, i.e., the number of times that template was found in the corpus, the third weight was the normalized count, i.e., the number of times the template appeared in the corpus divided by the number of times the input part was present in the corpus, finally, the fourth weight was a small constant (10^{-30}). The intention of this last weight was to ease the combination with the templates from the dictionary. For these, the first three weights were assigned the same small constant and the fourth was the probability of the translation of the pair obtained from the stochastic dictionary. This weighting schema allowed to separate the influence of the dictionary in smoothing the templates.

[1]They should have been probabilities, but in two of the cases there was no normalization and in one they were even greater than one!

Table 1: Statistics of the training corpora. The languages are German (De), English (En), Spanish (Es), Finnish (Fi) and French (Fr).

Languages	Sentences	Words (input/output)
De-En	751 088	15 257 871 / 16 052 702
Es-En	730 740	15 725 136 / 15 222 505
Fi-En	716 960	11 318 863 / 15 493 334
Fr-En	688 031	15 599 184 / 13 808 505

9 Experiments

In order to test the model, we have decided to participate in the shared task for this workshop.

9.1 The task

The aim of the task was to translate a set of 2,000 sentences from German, Spanish, Finnish and French into English. Those sentences were extracted from the Europarl corpus (Koehn, Unpublished). As training material, four different corpora were provided, one for each language pair, comprising around 700 000 sentence pairs each. Some details about these corpora can be seen in Table 1. An automatic alignment for each corpus was also provided.

The original sentence pairs were splitted using the techniques discussed in section 7.3. The total number of sentences after the split is presented in Table 2. Two different alignments were used: (a) the one provided in the definition of the task and (b) one obtained using GIZA++ (Och and Ney, 2003) to train an IBM's model 4. As it can be seen, the number of parts is very similar in both cases. The

Table 2: Number of training pairs after splitting to a maximum length of ten. "Provided" refers to the alignment provided in the task, "GIZA++" to those obtained with GIZA++.

	Sentence pairs	
Languages	Provided	GIZA++
De-En	2 351 121	2 282 316
Es-En	2 160 039	2 137 301
Fi-En	2 099 634	2 017 130
Fr-En	2 112 931	2 080 200

Table 3: Number of templates for each language pair: "Alignment" shows the number of templates derived from the alignments; "dictionary", those obtained from the dictionary; and "total" is the sum.

(a) Using the alignments provided with the task.

Lang.	Alignment	Dictionary	Total
De-En	2 660 745	1 840 582	4 501 327
Es-En	2 241 344	1 385 086	3 626 430
Fi-En	2 830 433	2 852 583	5 683 016
Fr-En	2 178 890	1 222 266	3 401 156

(b) Using GIZA++.

Lang.	Alignment	Dictionary	Total
De-En	2 672 079	1 796 887	4 468 966
Es-En	2 220 533	1 350 526	3 571 059
Fi-En	2 823 769	2 769 929	5 593 698
Fr-En	2 140 041	1 181 990	3 322 031

number of pairs after splitting is roughly three times the original.

Templates were extracted as described in section 8.1. The number of templates we obtained can be seen in Table 3. Again, the influence of the type of alignment was small. Except for Finnish, the number of dictionary templates was roughly two thirds of the templates extracted from the alignments.

9.2 Obtaining the translations

Once the templates were obtained, the development corpora were used to search for adequate values of the weights that `pharaoh` uses for each template (these are the weights passed to option `weight-t`, the other weights were not changed as an initial exploration seemed to indicate that they had little impact). As expected, the best weights differed between language pairs. The values can be seen in table 4.

Table 4: Best weights for each language pair. The columns are for the probability given by the model, the counts of the templates, the normalized counts and the weight given to the dictionary.

(a) Using the alignments provided with the task.

Languages	Model	Count	Norm	Dict
De-En	0.0	3.0	0.0	0.3
Es-En	0.0	2.9	0.0	0.4
Fi-En	0.0	7.0	0.0	0.0
Fr-En	0.0	7.0	1.0	1.0

(b) Using GIZA++.

Languages	Model	Count	Norm	Dict
De-En	0.0	3.0	0.0	0.0
Es-En	0.0	2.9	0.0	0.4
Fi-En	0.0	3.0	1.5	0.0
Fr-En	0.0	3.0	1.0	0.4

Table 5: BLEU scores of the translations.

	BLEU	
Languages	Provided	GIZA++
De-En	18.08	18.89
Es-En	21.65	21.48
Fi-En	13.31	13.79
Fr-En	21.25	19.86

It is interesting to note that the probabilities assigned by the model to the templates seemed to be better not taken into account. The most important feature was the counts of the templates, which sometimes were helped by the use of the dictionary, although that effect was small. Normalization of counts also had little impact.

10 Results and discussion

The results over the test sets can be seen in Table 5. It can be seen that, except for French, the influence of the initial alignment is very small. Also, the best results are obtained for Spanish and French, which are more similar to English that German or Finnish.

There are still many open questions that deserve more experimentation. The first is the influence of the split of the original corpora. Although the similarity of results seem to indicate that it has little influence, this has to be tested. Two more relevant aspects are whether the weighting schema is the best for the decoder. In particular, it is surprising that the normalization of counts had so little effect.

Finally, the average number of words per template is below two, which probably is too low. It is interesting to find alternate ways of obtaining the templates, for instance using internal nodes up to a given height or covering portions of the sentences up to a predefined number of words.

11 Conclusions

A new translation model has been presented. This model produces translations in a recursive way: the input sentence is divided in two parts, each is translated using the same procedure recursively and the translations are concatenated. The model has been used for finding the templates in a large vocabulary translation task. This involved using several heuristics to improve training time, including a method for splitting the input before training the models. Finally, the influence of using a stochastic dictionary together with the templates as a means of smoothing has been explored.

References

Leonard E. Baum and J. A. Eagon. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Ido Dagan, Kenneth W. Church, and William A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, Columbus, Ohio (USA). ACL.

Yonggang Deng, Shankar Kumar, and William Byrne. 2004. Bitext chunk alignment for statistical machine translation. Research Note 50, CLSP Johns Hopkins University, April.

P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. 1991. An inequality for rational functions with applications to some statistical problems. *IEEE Transactions on Information Theory*, 37(1):107–113, January.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124.

Philipp Koehn. Unpublished. Europarl: A multilingual corpus for evaluation of machine translation. Draft.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Joseph Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, December.

Enrique Vidal, Roberto Pieraccini, and Esther Levin. 1993. Learning associations between grammars: A new approach to natural language understanding. In *Proceedings of the EuroSpeech'93*, pages 1187–1190, Berlin (Germany).

Juan Miguel Vilar Torres. 1998. *Aprendizaje de Traductores Subsecuenciales para su empleo en tareas de dominio restringido*. Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia (Spain). (in Spanish).

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the COLING'96*, pages 836–841, Copenhagen (Denmark), August.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Richard Zens and Hermann Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 144–151, Sapporo (Japan), July. Association for Computational Lingustics.

Training and Evaluating Error Minimization Rules for Statistical Machine Translation

Ashish Venugopal
School of Computer Science
Carnegie Mellon University
arv@andrew.cmu.edu

Andreas Zollmann
School of Computer Science
Carnegie Mellon University
zollmann@cs.cmu.edu

Alex Waibel
School of Computer Science
Carnegie Mellon University
waibel@cs.cmu.edu

Abstract

Decision rules that explicitly account for non-probabilistic evaluation metrics in machine translation typically require special training, often to estimate parameters in exponential models that govern the search space and the selection of candidate translations. While the traditional Maximum A Posteriori (MAP) decision rule can be optimized as a piecewise linear function in a greedy search of the parameter space, the Minimum Bayes Risk (MBR) decision rule is not well suited to this technique, a condition that makes past results difficult to compare. We present a novel training approach for non-tractable decision rules, allowing us to compare and evaluate these and other decision rules on a large scale translation task, taking advantage of the high dimensional parameter space available to the phrase based Pharaoh decoder. This comparison is timely, and important, as decoders evolve to represent more complex search space decisions and are evaluated against innovative evaluation metrics of translation quality.

1 Introduction

State of the art statistical machine translation takes advantage of exponential models to incorporate a large set of potentially overlapping features to select translations from a set of potential candidates. As discussed in (Och, 2003), the direct translation model represents the probability of target sentence 'English' $\mathbf{e} = \mathbf{e_1} \ldots \mathbf{e_I}$ being the translation for a source sentence 'French' $\mathbf{f} = \mathbf{f_1} \ldots \mathbf{f_J}$ through an *exponential*, or *log-linear* model

$$p_\lambda(\mathbf{e}|\mathbf{f}) = \frac{\exp(\sum_{k=1}^{m} \lambda_k * h_k(\mathbf{e}, \mathbf{f}))}{\sum_{\mathbf{e}' \in \mathbf{E}} \exp(\sum_{k=1}^{m} \lambda_k * h_k(\mathbf{e}', \mathbf{f}))} \quad (1)$$

where $\mathbf{e}$ is a single candidate translation for $\mathbf{f}$ from the set of all English translations $\mathbf{E}$, λ is the parameter vector for the model, and each h_k is a feature function of $\mathbf{e}$ and $\mathbf{f}$. In practice, we restrict $\mathbf{E}$ to the set *Gen*($\mathbf{f}$) which is a set of highly likely translations discovered by a decoder (Vogel et al., 2003). Selecting a translation from this model under the Maximum A Posteriori (*MAP*) criteria yields

$$\text{transl}_\lambda(\mathbf{f}) = \arg\max_{\mathbf{e}} p_\lambda(\mathbf{e}|\mathbf{f}) \; . \quad (2)$$

This decision rule is optimal under the zero-one loss function, minimizing the Sentence Error Rate (Mangu et al., 2000). Using the log-linear form to model $p_\lambda(\mathbf{e}|\mathbf{f})$ gives us the flexibility to introduce overlapping features that can represent global context while decoding (searching the space of candidate translations) and rescoring (ranking a set of candidate translations before performing the $\arg\max$ operation), albeit at the cost of the traditional source-channel generative model of translation proposed in (Brown et al., 1993).

A significant impact of this paradigm shift, however, has been the movement to leverage the flexibility of the exponential model to maximize performance with respect to automatic evaluation met-

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 208–215,
Ann Arbor, June 2005.

rics. Each evaluation metric considers different aspects of translation quality, both at the sentence and corpus level, often achieving high correlation to human evaluation (Doddington, 2002). It is clear that the decision rule stated in (1) does not reflect the choice of evaluation metric, and substantial work has been done to correct this mismatch in criteria. Approaches include integrating the metric into the decision rule, and learning λ to optimize the performance of the decision rule. In this paper we will compare and evaluate several aspects of these techniques, focusing on Minimum Error Rate (MER) training (Och, 2003) and Minimum Bayes Risk (MBR) decision rules, within a novel training environment that isolates the impact of each component of these methods.

2 Addressing Evaluation Metrics

We now describe competing strategies to address the problem of modeling the evaluation metric within the decoding and rescoring process, and introduce our contribution towards training non-tractable error surfaces. The methods discussed below make use of *Gen*($\mathbf{f}$), the approximation to the complete candidate translation space $\mathbf{E}$, referred to as an *n-best list*. Details regarding *n-best list* generation from decoder output can be found in (Ueffing et al., 2002).

2.1 Minimum Error Rate Training

The predominant approach to reconciling the mismatch between the MAP decision rule and the evaluation metric has been to train the parameters λ of the exponential model to correlate the *MAP* choice with the maximum score as indicated by the evaluation metric on a development set with known references (Och, 2003). We differentiate between the decision rule

$$\mathrm{transl}_\lambda(\mathbf{f}) = \arg\max_{\mathbf{e} \in \mathit{Gen}(\mathbf{f})} p_\lambda(\mathbf{e}|\mathbf{f}) \qquad (3a)$$

and the training criterion

$$\hat{\lambda} = \arg\min_{\lambda} \mathit{Loss}(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}}) \qquad (3b)$$

where the Loss function returns an evaluation result quantifying the difference between the English candidate translation $\mathrm{transl}_\lambda(\mathbf{f})$ and its corresponding reference $\mathbf{r}$ for a source sentence $\mathbf{f}$. We indicate that this loss function is operating on a sequence of sentences with the vector notation. To avoid overfitting, and since MT researchers are generally blessed with an abundance of data, these sentences are from a separate development set.

The optimization problem (3b) is hard since the $\arg\max$ of (3a) causes the error surface to change in steps in $\mathbb{R}^m$, precluding the use of gradient based optimization methods. Smoothed error counts can be used to approximate the $\arg\max$ operator, but the resulting function still contains local minima. Grid-based line search approaches like Powell's algorithm could be applied but we can expect difficultly when choosing the appropriate grid size and starting parameters. In the following, we summarize the optimization algorithm for the unsmoothed error counts presented in (Och, 2003) and the implementation detailed in (Venugopal and Vogel, 2005).

- Regard $\mathit{Loss}(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}})$ as defined in (3b) as a function of the parameter vector λ to optimize and take the $\arg\max$ to compute $\mathrm{transl}_\lambda(\vec{\mathbf{f}})$ over the translations *Gen*($\mathbf{f}$) according to the *n-best* list generated with an initial estimate λ^0.

- The error surface defined by Loss (as a function of λ) is piecewise linear with respect to a single model parameter λ_k, hence we can determine exactly where it would be useful (values that change the result of the $\arg\max$) to evaluate λ_k for a given sentence using a simple line intersection method.

- Merge the list of useful evaluation points for λ_k and evaluate the corpus level $\mathit{Loss}(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}})$ at each one.

- Select the model parameter that represents the lowest Loss as k varies, set λ_k and consider the parameter λ_j for another dimension j.

This training algorithm, referred to as minimum error rate (MER) training, is a greedy search in each dimension of λ, made efficient by realizing that within each dimension, we can compute the points at which changes in λ actually have an impact on Loss. The appropriate considerations for termination and initial starting points relevant to any greedy search procedure must be accounted for. From the

nature of the training procedure and the *MAP* decision rule, we can expect that the parameters selected by MER training will strongly favor a few translations in the *n-best* list, namely for each source sentence the one resulting in the best score, moving most of the probability mass towards the translation that it believes should be selected. This is due to the decision rule, rather than the training procedure, as we will see when we consider alternative decision rules.

2.2 The Minimum Bayes Risk Decision Rule

The Minimum Bayes Risk Decision Rule as proposed by (Mangu et al., 2000) for the Word Error Rate Metric in speech recognition, and (Kumar and Byrne, 2004) when applied to translation, changes the decision rule in (2) to select the translation that has the lowest expected loss $\mathbf{E}[Loss(\mathbf{e}, \mathbf{r})]$, which can be estimated by considering a weighted $Loss$ between $\mathbf{e}$ and the elements of the *n-best* list, the approximation to $\mathbf{E}$, as described in (Mangu et al., 2000). The resulting decision rule is:

$$\mathrm{transl}_\lambda(\mathbf{f}) = \arg\min_{\mathbf{e} \in Gen(\mathbf{f})} \sum_{\mathbf{e}' \in Gen(\mathbf{f})} Loss(\mathbf{e}, \mathbf{e}') p_\lambda(\mathbf{e}'|\mathbf{f}) \ . \tag{4}$$

(Kumar and Byrne, 2004) explicitly consider selecting both $\mathbf{e}$ and $\mathbf{a}$, an alignment between the English and French sentences. Under a phrase based translation model (Koehn et al., 2003; Marcu and Wong, 2002), this distinction is important and will be discussed in more detail. The representation of the evaluation metric or the $Loss$ function is in the decision rule, rather than in the training criterion for the exponential model. This criterion is hard to optimize for the same reason as the criterion in (3b): the objective function is not continuous in λ. To make things worse, it is more expensive to evaluate the function at a given λ, since the decision rule involves a sum over all translations.

2.3 MBR and the Exponential Model

Previous work has reported the success of the MBR decision rule with fixed parameters relating independent underlying models, typically including only the language model and the translation model as features in the exponential model.

We extend the MBR approach by developing a training method to optimize the parameters λ in the exponential model as an explicit form for the conditional distribution in equation (1). The training task under the MBR criterion is

$$\lambda^* = \arg\min_\lambda Loss(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}}) \tag{5a}$$

where

$$\mathrm{transl}_\lambda(\mathbf{f}) = \arg\min_{\mathbf{e} \in Gen(\mathbf{f})} \sum_{\mathbf{e}' \in Gen(\mathbf{f})} Loss(\mathbf{e}, \mathbf{e}') p_\lambda(\mathbf{e}'|\mathbf{f}) \ . \tag{5b}$$

We begin with several observations about this optimization criterion.

- The MAP optimal λ^* are not the optimal parameters for this training criterion.
- We can expect the error surface of the MBR training criterion to contain larger sections of similar altitude, since the decision rule emphasizes consensus.
- The piecewise linearity observation made in (Papineni et al., 2002) is no longer applicable since we cannot move the log operation into the expected value.

3 Score Sampling

Motivated by the challenges that the MBR training criterion presents, we present a training method that is based on the assumption that the error surface is locally non-smooth but consists of local regions of similar $Loss$ values. We would like to focus the search within regions of the parameter space that result in low $Loss$ values, simulating the effect that the MER training process achieves when it determines the merged error boundaries across a set of sentences.

Let $Score(\lambda)$ be some function of $Loss(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}})$ that is greater or equal zero, decreases monotonically with $Loss$, and for which $\int (Score(\lambda) - \min_{\lambda'} Score(\lambda')) d\lambda$ is finite; e.g., $1 - Loss(\mathrm{transl}_\lambda(\vec{\mathbf{f}}), \vec{\mathbf{r}})$ for the word-error rate (WER) loss and a bounded parameter space. While sampling parameter vectors λ and estimating $Loss$ in these points, we will constantly refine our estimate of the error surface and thereby of the $Score$ function. The main idea in our score

sampling algorithm is to make use of this $Score$ estimate by constructing a probability distribution over the parameter space that depends on the $Score$ estimate in the current iteration step i and sample the parameter vector λ^{i+1} for the next iteration from that distribution. More precisely, let $\widehat{Sc}^{(i)}$ be the estimate of $Score$ in iteration i (we will explain how to obtain this estimate below). Then the probability distribution from which we sample the parameter vector to test in the next iteration is given by:

$$p(\lambda) = \frac{\widehat{Sc}^{(i)}(\lambda) - \min_{\lambda'} \widehat{Sc}^{(i)}(\lambda')}{\int (\widehat{Sc}^{(i)}(\lambda) - \min_{\lambda'} \widehat{Sc}^{(i)}(\lambda'))\, d\lambda} \quad (6)$$

This distribution produces a sequence $\lambda^1, \ldots, \lambda^n$ of parameter vectors that are more concentrated in areas that result in a high $Score$. We can select the value from this sequence that generates the highest $Score$, just as in the MER training process.

The exact method of obtaining the $Score$ estimate $\widehat{Sc}$ is crucial: If we are not careful enough and guess too low values of $\widehat{Sc}(\lambda)$ for parameter regions that are still unknown to us, the resulting sampling distribution p might be zero in those regions and thus potentially optimal parameters might never be sampled. Rather than aiming for a consistent estimator of $Score$ (i.e., an estimator that converges to $Score$ when the sample size goes to infinity), we design $\widehat{Sc}$ with regard to yielding a suitable sampling distribution p.

Assume that the parameter space is bounded such that $min_k \leq \lambda_k \leq max_k$ for each dimension k, We then define a set of pivots $\mathcal{P}$, forming a grid of points in $\mathbb{R}^m$ that are evenly spaced between min_k and max_k for each dimension k. Each pivot represents a region of the parameter space where we expect generally consistent values of $Score$. We do not restrict the values of λ_m to be at these pivot points as a grid search would do, rather we treat the pivots as landmarks within the search space.

We approximate the distribution $p(\lambda)$ with the discrete distribution $p(\lambda \in \mathcal{P})$, leaving the problem of estimating $|\mathcal{P}|$ parameters. Initially, we set p to be uniform, i.e., $p^{(0)}(\lambda) = 1/|\mathcal{P}|$. For subsequent iterations, we now need an estimate of $Score(\lambda)$ for each pivot $\lambda \in \mathcal{P}$ in the discrete version of equation (6) to obtain the new sampling distribution p. Each iteration i proceeds as follows.

- Sample $\widetilde{\lambda^i}$ from the discrete distribution $p^{(i-1)}(\lambda \in \mathcal{P})$ obtained by the previous iteration.
- Sample the new parameter vector λ^i by choosing for each $k \in \{1, \ldots, m\}$, $\lambda^i_k := \widetilde{\lambda^i_k} + \varepsilon_k$, where ε_k is sampled uniformly from the interval $(-d_k/2, d_k/2)$ and d_k is the distance between neighboring pivot points along dimension k. Thus, λ^i is sampled from a region around the sampled pivot.
- Evaluate $Score(\lambda^i)$ and distribute this score to obtain new estimates $\widehat{Sc}^{(i)}(\lambda)$ for all pivots $\lambda \in \mathcal{P}$ as described below.
- Use the updated estimates $\widehat{Sc}^{(i)}$ to generate the sampling distribution $p^{(i)}$ for the next iteration according to

$$p^{(i)}(\lambda) = \frac{\widehat{Sc}^{(i)}(\lambda) - \min_{\lambda'} \widehat{Sc}^{(i)}(\lambda')}{\sum_{\lambda \in \mathcal{P}} (\widehat{Sc}^{(i)}(\lambda) - \min_{\lambda'} \widehat{Sc}^{(i)}(\lambda'))} .$$

The score $Score(\lambda^i)$ of the currently evaluated parameter vector does not only influence the score estimate at the pivot point of the respective region, but the estimates at all pivot points. The closest pivots are influenced most strongly. More precisely, for each pivot $\lambda \in \mathcal{P}$, $\widehat{Sc}^{(i)}(\lambda)$ is a weighted average of $Score(\lambda^1), \ldots, Score(\lambda^i)$, where the weights $w^{(i)}(\lambda)$ are chosen according to

$$\begin{aligned} w^{(i)}(\lambda) &= \mathrm{infl}^{(i)}(\lambda) \times \mathrm{corr}^{(i)}(\lambda) \quad \text{with} \\ \mathrm{infl}^{(i)}(\lambda) &= \mathrm{mvnpdf}(\lambda, \lambda^i, \Sigma) \quad \text{and} \\ \mathrm{corr}^{(i)}(\lambda) &= 1/p^{(i-1)}(\lambda) . \end{aligned}$$

Here, $\mathrm{mvnpdf}(x, \mu, \Sigma)$ denotes the m-dimensional multivariate-normal probability density function with mean μ and covariance matrix Σ, evaluated at point x. We chose the covariance matrix $\Sigma = \mathrm{diag}(d_1^2, \ldots, d_m^2)$, where again d_k is the distance between neighboring grid points along dimension k. The term $\mathrm{infl}^{(i)}(\lambda)$ quantifies the influence of the evaluated point λ^i on the pivot λ, while $\mathrm{corr}^{(i)}(\lambda)$ is a correction term for the bias introduced by having sampled λ^i from $p^{(i-1)}$.

Smoothing uncertain regions In the beginning of the optimization process, there will be pivot regions that have not yet been sampled from and for which not even close-by regions have been sampled yet. This will be reflected in the low sum of influence terms $\text{infl}^{(1)}(\lambda) + \cdots + \text{infl}^{(i)}(\lambda)$ of the respective pivot points λ. It is therefore advisable to discount some probability mass from $p^{(i)}$ and distribute it over pivots with low influence sums (reflecting low confidence in the respective score estimates) according to some smoothing procedure.

4 N-Best lists in Phrase Based Decoding

The methods described above make extensive use of *n-best* lists to approximate the search space of candidate translations. In phrase based decoding we often interpret the MAP decision rule to select the top scoring path in the translation lattice. Selecting a particular path means in fact selecting the pair $\langle \mathbf{e}, \mathbf{s} \rangle$, where $\mathbf{s}$ is a segmentation of the the source sentence $\mathbf{f}$ into phrases and alignments onto their translations in $\mathbf{e}$. Kumar and Byrne (2004) represent this decision explicitly, since the $Loss$ metrics considered in their work evaluate alignment information as well as lexical (word) level output. When considering lexical scores as we do here, the decision rule minimizing 0/1 loss actually needs to take the sum over all potential segmentations that can generate the same word sequence. In practice, we only consider the high probability segmentation decisions, namely the ones that were found in the *n-best* list. This gives the *0/1 loss* criterion shown below.

$$\text{transl}_\lambda(\mathbf{f}) = \arg\max_{\mathbf{e}} \sum_{\mathbf{s}} p_\lambda(\mathbf{e}, \mathbf{s}|\mathbf{f}) \qquad (7)$$

The *0/1 loss* criterion favors translations that are supported by several segmentation decisions. In the context of phrase-based translations, this is a useful criterion, since a given lexical target word sequence can be correctly segmented in several different ways, all of which would be scored equally by an evaluation metric that only considers the word sequence.

5 Experimental Framework

Our goal is to evaluate the impact of the three decision rules discussed above on a large scale translation task that takes advantage of multidimensional features in the exponential model. In this section we describe the experimental framework used in this evaluation.

5.1 Data Sets and Resources

We perform our analysis on the data provided by the 2005 ACL Workshop in Exploiting Parallel Texts for Statistical Machine Translation, working with the French-English Europarl corpus. This corpus consists of 688031 sentence pairs, with approximately 156 million words on the French side, and 138 million words on the English side. We use the data as provided by the workshop and run lower casing as our only preprocessing step. We use the 15.5 million entry phrase translation table as provided for the shared workshop task for the French-English data set. Each translation pair has a set of 5 associated phrase translation scores that represent the maximum likelihood estimate of the phrase as well as internal alignment probabilities. We also use the English language model as provided for the shared task. Since each of these decision rules has its respective training process, we split the workshop test set of 2000 sentences into a development and test set using random splitting. We tried two decoders for translating these sets. The first system is the Pharaoh decoder provided by (Koehn et al., 2003) for the shared data task. The Pharaoh decoder has support for multiple translation and language model scores as well as simple phrase distortion and word length models. The pruning and distortion limit parameters remain the same as in the provided initialization scripts, i.e., $DistortionLimit = 4, BeamThreshold = 0.1, Stack = 100$. For further information on these parameter settings, confer (Koehn et al., 2003). Pharaoh is interesting for our optimization task because its eight different models lead to a search space with seven free parameters. Here, a principled optimization procedure is crucial. The second decoder we tried is the CMU Statistical Translation System (Vogel et al., 2003) augmented with the four translation models provided by the Pharaoh system, in the following called CMU-Pharaoh. This system also leads to a search space with seven free parameters.

5.2 N-Best lists

As mentioned earlier, the model parameters λ play a large role in the search space explored by a pruning beam search decoder. These parameters affect the histogram and beam pruning as well as the future cost estimation used in the Pharaoh and CMU decoders. The initial parameter file for Pharaoh provided by the workshop provided a very poor estimate of λ, resulting in an n-best list of limited potential. To account for this condition, we ran Minimum Error Rate training on the development data to determine scaling factors that can generate a *n-best* list with high quality translations. We realize that this step biases the *n-best* list towards the MAP criteria, since its parameters will likely cause more aggressive pruning. However, since we have chosen a large N=1000, and retrain the MBR, MAP, and 0/1 loss parameters separately, we do not feel that the bias has a strong impact on the evaluation.

5.3 Evaluation Metric

This paper focuses on the BLEU metric as presented in (Papineni et al., 2002). The BLEU metric is defined on a corpus level as follows.

$$Score(\vec{\mathbf{e}}, \vec{\mathbf{r}}) = BP(\vec{\mathbf{e}}, \vec{\mathbf{r}}) * \exp(\frac{1}{N}\sum_{1}^{N}(\log p_n))$$

where p_n represent the precision of n-grams suggested in $\vec{\mathbf{e}}$ and BP is a brevity penalty measuring the relative shortness of $\vec{\mathbf{e}}$ over the whole corpus. To use the BLEU metric in the candidate pairwise loss calculation in (4), we need to make a decision regarding cases where higher order n-grams matches are not found between two candidates. Kumar and Byrne (2004) suggest that if any n-grams are not matched then the pairwise BLEU score is set to zero. As an alternative we first estimate corpus-wide n-gram counts on the development set. When the pairwise counts are collected between sentences pairs, they are added onto the baseline corpus counts to and scored by BLEU. This scoring simulates the process of scoring additional sentences after seeing a whole corpus.

5.4 Training Environment

It is important to separate the impact of the decision rule from the success of the training procedure. To appropriately compare the MAP, 0/1 loss and MBR decisions rules, they must all be trained with the same training method, here we use the Score Sampling training method described above. We also report MAP scores using the MER training described above to determine the impact of the training algorithm for MAP. Note that the MER training approach cannot be performed on the MBR decision rule, as explained in Section 2.3. MER training is initialized at random values of λ and run (successive greedy search over the parameters) until there is no change in the error for three complete cycles through the parameter set. This process is repeated with new starting parameters as well as permutations of the parameter search order to ensure that there is no bias in the search towards a particular parameter. To improve efficiency, pairwise scores are cached across requests for the score at different values of λ, and for MBR only the $\mathbf{E}[Loss(\mathbf{e}, \mathbf{r})]$ for the top twenty hypotheses as ranked by the model are computed.

6 Results

The results in Table 1 compare the BLEU score achieved by each training method on the development and test data for both Pharaoh and CMU-Pharaoh. Score-sampling training was run for 150 iterations to find λ for each decision rule. The MAP-MER training was performed to evaluate the effect of the greedy search method on the generalization of the development set results. Each row represents an alternative training method described in this paper, while the test set columns indicate the criteria used to select the final translation output $\vec{\mathbf{e}}$. The bold face scores are the scores for matching training and testing methods. The underlined score is the highest test set score, achieved by MBR decoding using the CMU-Pharaoh system trained for the MBR decision rule with the score-sampling algorithm. When comparing MER training for MAP-decoding with score-sampling training for MAP-decoding, score-sampling surprisingly outperforms MER training for both Pharaoh and CMU-Pharaoh, although MER training is specifically tailored to the MAP metric. Note, however, that our score-sampling algorithm has a considerably longer running time (several hours) than the MER algorithm (several minutes). Interestingly, within MER train-

training method	Dev. set sc.	test set sc. MAP	test set sc. 0/1 loss	test set sc. MBR
MAP MER (Pharaoh)	29.08	**29.30**	29.42	29.36
MAP score-sampl. (Pharaoh)	29.08	**29.41**	29.24	29.30
0/1 loss sc.-s. (Pharaoh)	29.08	29.16	**29.28**	29.30
MBR sc.-s. (Pharaoh)	29.00	29.11	29.08	**29.17**
MAP MER (CMU-Pharaoh)	28.80	**29.02**	29.41	29.60
MAP sc.-s. (CMU-Ph.)	29.10	**29.85**	29.75	29.55
0/1 loss sc.-s. (CMU-Ph.)	28.36	29.97	**29.91**	29.72
MBR sc.-s. (CMU-Ph.)	28.36	30.18	30.16	**30.28**

Table 1. Comparing BLEU scores generated by alternative training methods and decision rules

ing for Pharaoh, the 0/1 loss metric is the top performer; we believe the reason for this disparity between training and test methods is the impact of phrasal consistency as a valuable measure within the *n-best* list.

The relative performance of MBR score-sampling w.r.t. MAP and 0/1-loss score sampling is quite different between Pharaoh and CMU-Pharaoh: While MBR score-sampling performs worse than MAP and 0/1-loss score sampling for Pharaoh, it yields the best test scores across the board for CMU-Pharaoh. A possible reason is that the n-best lists generated by Pharaoh have a large percentage of lexically identical translations, differing only in their segmentations. As a result, the 1000-best lists generated by Pharaoh contain only a small percentage of unique translations, a condition that reduces the potential of the Minimum Bayes Risk methods. The CMU decoder, contrariwise, prunes away alternatives below a certain score-threshold during decoding and does not recover them when generating the n-best list. The n-best lists of this system are therefore typically more diverse and in particular contain far more unique translations.

7 Conclusions and Further Work

This work describes a general algorithm for the efficient optimization of error counts for an arbitrary $Loss$ function, allowing us to compare and evaluate the impact of alternative decision rules for statistical machine translation. Our results suggest the value and sensitivity of the translation process to the $Loss$ function at the decoding and reordering stages of the process. As phrase-based translation and reordering models begin to dominate the state of the art in machine translation, it will become increasingly important to understand the nature and consistency of *n-best* list training approaches. Our results are reported on a complete package of translation tools and resources, allowing the reader to easily recreate and build upon our framework. Further research might lie in finding efficient representations of Bayes Risk loss functions within the decoding process (rather than just using MBR to rescore n-best lists), as well as analyses on different language pairs from the available Europarl data. We have shown score sampling to be an effective training method to conduct these experiments and we hope to establish its use in the changing landscape of automatic translation evaluation. The source code is available at: `www.cs.cmu.edu/~zollmann/scoresampling/`

8 Acknowledgments

We thank Stephan Vogel, Ying Zhang, and the anonymous reviewers for their valuable comments and suggestions.

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *In Proc. ARPA Workshop on Human Language Technology*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North*

American Association for Computational Linguistics Conference (HLT/NAACL), Edomonton, Canada, May 27-June 1.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Boston,MA, May 27-June 1.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *CoRR*, cs.CL/0010012.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadephia, PA, July 6-7.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics*, pages 311–318.

Nicola Ueffing, Franz Josef Och, and Hermann Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadephia, PA, July 6-7.

Ashish Venugopal and Stephan Vogel. 2005. Considerations in mce and mmi training for statistical machine translation. In *Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05)*, Budapest, Hungary, May. The European Association for Machine Translation.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical translation system. In *Proceedings of MT Summit IX*, New Orleans, LA, September.

Author Index