Guiding Statistical Word Alignment Models With Prior Knowledge

Yonggang Deng and **Yuqing Gao**
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{ydeng,yuqing}@us.ibm.com

Abstract

We present a general framework to incorporate prior knowledge such as heuristics or linguistic features in statistical generative word alignment models. Prior knowledge plays a role of probabilistic soft constraints between bilingual word pairs that shall be used to guide word alignment model training. We investigate knowledge that can be derived automatically from entropy principle and bilingual latent semantic analysis and show how they can be applied to improve translation performance.

1 Introduction

Statistical word alignment models learn word associations between parallel sentences from statistics. Most models are trained from corpora in an unsupervised manner whose success is heavily dependent on the quality and quantity of the training data. It has been shown that human knowledge, in the form of a small amount of manually annotated parallel data to be used to seed or guide model training, can significantly improve word alignment F-measure and translation performance (Ittycheriah and Roukos, 2005; Fraser and Marcu, 2006).

As formulated in the competitive linking algorithm (Melamed, 2000), the problem of word alignment can be regarded as a process of word linkage disambiguation, that is, choosing correct associations among all competing hypothesis. The more reasonable constraints are imposed on this process, the easier the task would become. For instance, the most relaxed IBM Model-1, which assumes that any source word can be generated by any target word equally regardless of distance, can be improved by demanding a Markov process of alignments as in HMM-based models (Vogel et al., 1996), or implementing a distribution of number of target words linked to a source word as in IBM fertility-based models (Brown et al., 1993).

Following the path, we shall put more constraints on word alignment models and investigate ways of implementing them in a statistical framework. We have seen examples showing that names tend to align to names and function words are likely to be linked to function words. These observations are independent of language and can be understood by common sense. Moreover, there are other linguistically motivated constraints. For instance, words aligned to each other presumably are semantically consistent; and likely to be, they are syntactically agreeable. In these paper, we shall exploit some of these constraints in building better word alignments in the application of statistical machine translation.

We propose a simple framework that can integrate prior knowledge into statistical word alignment model training. In the framework, prior knowledge serves as probabilistic soft constraints that will guide word alignment model training. We present two types of constraints that are derived in an unsupervised way: one is based on the entropy principle, the other comes from bilingual latent semantic analysis. We investigate their impact on word alignments and show their effectiveness in improving translation performance.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 1–8,
Prague, Czech Republic, June 2007.

2 Constrained Word Alignment Models

The framework that we propose to incorporate statistical constraints into word alignment models is generic. It can be applied to complicated models such IBM Model-4 (Brown et al., 1993). We shall take HMM-based word alignment model (Vogel et al., 1996) as an example and follow the notation of (Brown et al., 1993). Let $\mathbf{e} = e_1^l$ represent a source string and $\mathbf{f} = f_1^m$ a target string. The random variable $\mathbf{a} = a_1^m$ specifies the indices of source words that target words are aligned to.

In an HMM-based word alignment model, source words are treated as Markov states while target words are observations that are generated when jumping to states:

$$P(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{m} P(a_j|a_{j-1}, \mathbf{e}) t(f_j|e_{a_j})$$

Notice that a target word f is generated from a source state e by a simple lookup of the translation table, a.k.a., t-table $t(f|e)$, as depicted in (A) of Figure 1. To incorporate prior knowledge or impose constraints, we introduce two nodes E and F representing the hidden tags of the source word e and the target word f respectively, and organize the dependency structure as in (B) of Figure 1. Given this generative procedure, f will also depend on its tag F, which is determined probabilistically by the source tag E. The dependency from E to F functions as a soft constraint showing how the two hidden tags are agreeable to each other. Mathematically, the conditional distribution follows:

$$\begin{aligned} P(f|e) &= \sum_{E,F} P(f, E, F|e) \\ &= \sum_{E,F} P(E|e)P(F|E)P(f|e, F) \\ &= t(f|e) \cdot Con(f, e), \end{aligned} \quad (1)$$

where

$$Con(f, e) = \sum_{E,F} P(E|e)P(F|E)P(F|f)/P(F) \quad (2)$$

is the soft weight attached to the t-table entry. It considers all possible hidden tags of e and f and serves as constraint between the link.

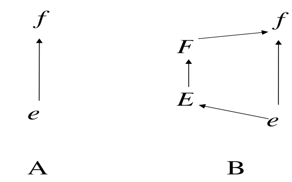

Figure 1: A simple table lookup (A) vs. a constrained procedure (B) of generating a target word f from a source word e.

We do not change the value of $Con(f, e)$ during iterative model training but rather keep it constant as an indicator of how strong the word pair should be considered as a candidate. This information is derived before word alignment model training and will act as soft constraints that need to be respected during training and alignments. For a given word pair, the soft constraint can have different assignment in different sentence pairs since the word tags can be context dependent.

To understand why we take the “detour” of generating a target word rather than directly from a t-table, consider the hidden tag as binary value indicating being a name or not. Without these constraints, t-table entries for names with low frequency tend to be flat and word alignments can be chosen randomly without sufficient statistics or strong lexical preference under maximum likelihood criterion. If we assume that a name is produced by a name with a high probability but by a non-name with a low probability, i.e. $P(F = E) >> P(F \neq E)$, proper names with low counts then are encouraged to link to proper names during training; and consequently, conditional probability mass would be more focused on correct name translations. On the other hand, names are discouraged to produce non-names. This will potentially avoid incorrect word associations. We are able to apply this type of constraint since usually there are many monolingual resources available to build a high performance probabilistic name tagger. The example suggests that putting reasonable constraints learned from monolingual analysis can alleviate data spareness problem in bilingual applications.

The weights $Con(f, e)$ are the prior knowledge that shall be assigned with care but respected during training. The baseline is to set all these weights to 1, which is equivalent to placing no prior knowledge on model training. The introduction of these weights does not complicate parameter estimation procedure. Whenever a source word e is hypothesized to generate a target word f, the translation probability $t(f|e)$ should be weighted by $Con(f, e)$.

We point out that the constraints between f and e through their hidden tags are in probabilities. There are no hard decisions made before training. A strong preference between two words can be expressed by assigning corresponding weights close to 1. This will affect the final alignment model.

Depending on the hidden tags, there are many realizations of reasonable constraints that can be put beforehand. They can be semantic classes, syntactic annotations, or as simple as whether being a function word or content word. Moreover, the source side and the target side do not have to share the same set of tags. The framework is also flexible to support multiple types of constraints that can be implemented in parallel or cascaded sequence. Moreover, the constraints between words can be dependent on context within parallel sentences. Next, we will describe two types of constraints that we proposed. Both of them are derived from data in an unsupervised way.

2.1 Entropy Principle

It is assumed that generally speaking, a source function word generates a target function word with a higher probability than generating a target content word; similar assumption applies to a source content word as well. We capture this type of constraint by defining the hidden tag E and F as binary labels indicating being a content word or not. Based on the assumption, we design probabilistic relationship between the two hidden tags as:

$$P(E = F) = 1 - P(E \neq F) = \alpha,$$

where α is a scalar whose value is close to 1, say 0.9. The bigger α is, the tighter constraint we put on word pairs to be connected requiring the same type of label.

To determine the probability of a word being a function word, we apply the entropy principle. A function word, say "of","in" or "have", appears more frequently than a content word, say "journal" or "chemistry", in a document or sentence. We will approximate the probability of a word as a function word with the relative uncertainty of its being observed in a sentence.

More specifically, suppose we have N parallel sentences in the training corpus. For each word w_i[1], let c_{ij} be the number of word w_i observed in the j-th sentence pair, and let c_i be the total number of occurrences of w_i in the corpus. We define the relative entropy of word w_i as

$$\epsilon_{w_i} = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{ij}}{c_i} \log \frac{c_{ij}}{c_i}.$$

With the entropy of a word, the likelihood of word w being tagged as a function word is approximated with $w^{(1)} = \epsilon_w$ and being tagged as a content word with $w^{(0)} = 1 - \epsilon_w$.

We ignore the denominator in Equ. (2) and find the constraint under the entropy principle:

$$\begin{aligned} Con(f, e) \;=\; & \alpha(e^{(0)} f^{(0)} + e^{(1)} f^{(1)}) + \\ & (1 - \alpha)(e^{(1)} f^{(0)} + e^{(0)} f^{(1)}). \end{aligned}$$

As can be seen, the connection between two words is simulated with a binary symmetric channel. An example distribution of the constraint function is illustrated in Figure 2. A high value of α encourages connecting word pairs with comparable entropy; When $\alpha = 0.5$, $Con(f, e)$ is constant which corresponds to applying no prior constraint; When α is close to 0, the function plays opposite role on word alignment training where a high frequency word is pushed to associate with a low frequency word.

2.2 Bilingual Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the meaning of words by statistically analyzing word contextual usages in a collection of text. It provides a method by which to calculate the similarity of meaning of given words and documents. LSA has been successfully applied to information retrieval (Deerwester et al., 1990), statistical langauge modeling (Bellegarda, 2000) and etc.

[1]We prefix 'E_' to source words and 'F_' to target words to distinguish words that have the same spelling but are from different languages.

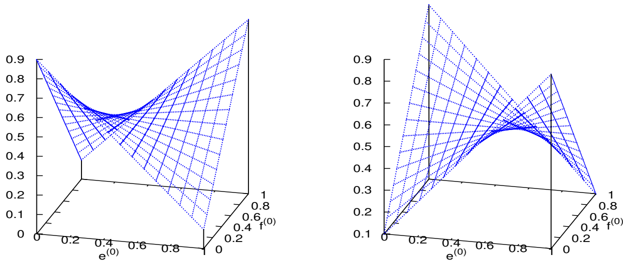

Figure 2: Distribution of the constraint function based on entropy principle when $\alpha = 0.9$ on the left and $\alpha = 0.1$ on the right.

We explore LSA techniques in bilingual environment to derive semantic constraints as prior knowledge for guiding a word alignment model training. The idea is to find semantic representation of source words and target words in the so-called low-dimensional LSA-space, and then to use their similarities to quantitatively establish semantic consistencies. We propose two different approaches.

2.2.1 A Simple Bag-of-word Model

One method we investigate is a simple bag-of-word model as in monolingual LSA. We treat each sentence pair as a document and do not distinguish source words and target words as if they are terms generated from the same vocabulary. A sparse matrix W characterizing word-document co-occurrence is constructed. Following the notation in section 2.1, the ij-th entry of the matrix W is defined as in (Bellegarda, 2000)

$$W_{ij} = (1 - \epsilon_{w_i})\frac{c_{ij}}{c_j},$$

where c_j is the total number of words in the j-th sentence pair. This construction considers the importance of words globally (corpus wide) and locally (within sentence pairs). Alternative constructions of the matrix are possible using raw counts or TF-IDF (Deerwester et al., 1990).

W is a $M \times N$ sparse matrix, where M is the size of vocabulary including both source and target words. To obtain a compact representation, singular value decomposition (SVD) is employed (cf. Berry et al (1993)) to yield $W \approx \hat{W} = U \times S \times V^T$ as Figure 3 shows, where, for some order $R \ll \min(M, N)$ of the decomposition, U is a $M \times R$ left singular matrix with rows u_i, $i = 1, \cdots, M$, S is a $R \times R$ diagonal matrix of singular values $s_1 \geq s_2 \geq \ldots \geq s_R \gg 0$, and V is $N \times R$ a right singular matrix with rows v_j, $j = 1, \cdots, N$. For each i, the scaled R-vector u_iS may be viewed as representing w_i, the i-th word in the vocabulary, and similarly the scaled R-vector v_jS as representing d_j, j-th document in the corpus. Note that the u_iS's and v_jS's both belong to $\mathbb{R}^R$, the so-called LSA-space. All target and source words are projected into the same LSA-space too.

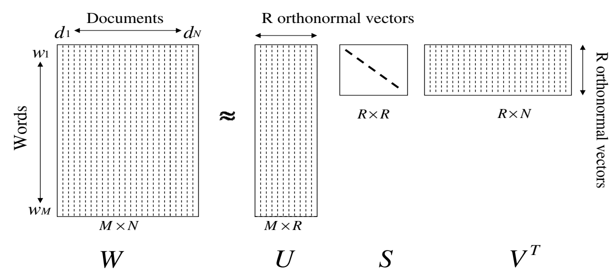

Figure 3: SVD of the Sparse Matrix W.

As Equ. (2) suggested, to induce semantic constraints in a straightforward way, one would proceed as follows: firstly, perform word semantic clustering with, say, their compact representations in the LSA-space; secondly, construct cluster generating dependencies by specifying the conditional distribution of $P(F|E)$; and finally, for each word pair, induce the semantic constraint by considering all possible semantic labeling schemes. We approximate this long process with simply finding word similarities defined by their cosine distance in the low dimension space:

$$Con(f, e) = \frac{1}{2}(\cos(u_fS, u_eS) + 1) \qquad (3)$$

The linear mapping above is introduced to avoid negative constraints and to set the maximum constraint value as 1.

In building word alignment models, a special "NULL" word is usually introduced to address target words that align to no source words. Since this physically non-existing word is not in the vocabulary of the bilingual LSA, we use the centroid of all source words as its vector representation in the LSA-space. The semantic constraints between "NULL" and any target words can be derived in the same way. However, this is chosen for mostly computational convenience, and is not the only way to address the empty word issue.

2.2.2 Utilizing Word Alignment Statistics

While the simple bag-of-word model puts all source words and target words as rows in the matrix, another method of deriving semantic constraint constructs the sparse matrix by taking source words as rows and target words as columns and uses statistics from word alignment training to form word pair co-occurrence association.

More specifically, we regard each target word f as a "document" and each source word e as a "term". The number of occurrences of the source word e in the document f is defined as the expected number of times that f generates e in the parallel corpus under the word alignment model. This method requires training the baseline word alignment model in another direction by taking fs as source words and es as target words, which is often done for symmetric alignments, and then dumping out the soft counts when model converges. We threshold the minimum word-to-word translation probability to remove word pairs that have low co-occurrence counts.

Following the similarity induced semantic constraints in section 2.2.1, we need to find the distance between a term and a document. Let v_f be the projection of the document representing the target word f and u_e the projection of the term representing the source word e after performing SVD on the sparse matrix, we calculate the similarity between (f, e) and then find their semantic constraint to be

$$Con(f, e) = \frac{1}{2}(\cos(v_f S^{1/2}, u_e S^{1/2}) + 1) \qquad (4)$$

Unlike the method in section 2.2.1, there is no empty word issue here since we do have statistics of the "NULL" word as a source word generating e words and therefore there is a "document" assigned to it.

3 Experimental Results

We test our framework on the task of large vocabulary translation from dialectical (Iraqi) Arabic utterances into English. The task covers multiple domains including travel, emergency medical diagnosis, defense-oriented force protection, security and etc. To avoid impacts of speech recognition errors, we only report experiments from text to text translation.

The training corpus consists of 390K sentence pairs, with total 2.43M Arabic words and 3.38M English words. These sentences are in typical spoken transcription form, i.e., spelling errors, disfluencies, such as word or phrase repetition, and ungrammatical utterances are commonly observed. Arabic utterance length ranges from 3 to 70 words with the average of 6 words.

There are 25K entries in the English vocabulary and 90K in Arabic side. Data sparseness severely challenges word alignment model and consequently automatic phrase translation induction. There are 42K singletons in Arabic vocabulary, and 14K Arabic words with occurrence of twice each in the corpus. Since Arabic is a morphologically rich language where affixes are attached to stem words to indicate gender, tense, case and etc, in order to reduce vocabulary size and address out-of-vocabulary words, we split Arabic words into affix and root according to a rule-based segmentation scheme (Xiang et al., 2006) with the help from the Buckwalter analyzer (LDC, 2002) output. This reduces the size of Arabic vocabulary to 52K.

Our test data consists of 1294 sentence pairs. They are split into two parts: half of them is used as the development set, on which training parameters and decoding feature weights are tuned, the other half is for test.

3.1 Training and Translation Setup

Starting from the collection of parallel training sentences, we train word alignment models in two translation directions, from English to Iraqi Arabic and from Iraqi Arabic to English, and derive two sets of Viterbi alignments. By combining word alignments in two directions using heuristics (Och and Ney, 2003), a single set of static word alignments is then formed. All phrase pairs which respect to the word alignment boundary constraint are identified and pooled to build phrase translation tables with the Maximum Likelihood criterion. We prune phrase translation entries by their probabilities. The maximum number of tokens in Arabic phrases is set to 5 for all conditions.

Our decoder is a phrase-based multi-stack implementation of the log-linear model similar to Pharaoh (Koehn et al., 2003). Like other log-linear model based decoders, active features in our translation engine include translation models in two directions, lexicon weights in two directions, language model, distortion model, and sentence length penalty. These feature weights are tuned on the dev set to achieve optimal translation performance using downhill simplex method (Och and Ney, 2002). The language model is a statistical trigram model estimated with Modified Kneser-Ney smoothing (Chen and Goodman, 1996) using all English sentences in the parallel training data.

We measure translation performance by the BLEU score (Papineni et al., 2002) and Translation Error Rate (TER) (Snover et al., 2006) with one reference for each hypothesis. Word alignment models trained with different constraints are compared to show their effects on the resulting phrase translation tables and the final translation performance.

3.2 Translation Results

Our baseline word alignment model is the word-to-word Hidden Markov Model (Vogel et al., 1996). Basic models in two translation directions are trained simultaneously where statistics of two directions are shared to learn symmetric translation lexicon and word alignments with high precision motivated by (Zens et al., 2004) and (Liang et al., 2006). The baseline translation results (BLEU and TER) on the dev and test set are presented in the line "HMM" of Table 1. We also compare with results of IBM Model-4 word alignments implemented in GIZA++ toolkit (Och and Ney, 2003).

We study and compare two types of constraint and see how they affect word alignments and translation output. One is based on the entropy principle as described in Section 2.1, where α is set to 0.9; The other is based on bilingual latent semantic analysis.

For the simple bag-of-word bilingual LSA as described in Section 2.2.1, after SVD on the sparse matrix using the toolkit SVDPACK (Berry et al., 1993), all source and target words are projected into a low-dimensional ($R = 88$) LSA-space. Word pair semantic constrains are calculated based on their similarity as in Equ. 3 before word alignment training. Like the baseline, we perform 6 iterations of IBM Model-1 training and then 4 iteration of HMM training. The semantic constraints are used to guide word alignment model training for each iteration. The BLEU score and TER with this constraint are shown in the line "BiLSA-1" of Table 1.

To exploit word alignment statistics in bilingual LSA as described in Section 2.2.2, we dump out the statistics of the baseline word alignment model and use them to construct the sparse matrix. We find low-dimensional representation ($R = 67$) of English words and Arabic words and use their similarity to establish semantic constraints as in Equ. 4. The training procedure is the same as the baseline and "BiLSA-1". The translation results with these word alignments are shown as "BiLSA-2" in Table 1.

As Table 1 shows, when the entropy based constraints are applied, BLEU score improves 0.5 point on the test set. Clearly, when bilingual LSA constraints are applied, translation performance can be improved up to 1.6 BLEU points. We also observe that TER can drop 2.1 points with the "BiLSA-1" constraint.

While "BiLSA-1" constraint performs better on the test set, "BiLSA-2" constraint achieves slightly higher BLEU score on the dev set. We then try a simple combination of these two types of constraints, that is the geometric mean of $Con_{BiLSA-1}(f, e)$ and $Con_{BiLSA-2}(f, e)$, and find out that BLEU score can be improved a little bit further on both sets as the line "Mix" shows.

We notice that the relatively simpler HMM model can perform comparable or better than the sophisticated Model-4 when proper constraints are active in guiding word alignment model training. We also try to put constraints in Model-4. As the Equation 1 implies, when a word-to-word generative probability is needed, one should multiply corresponding lexicon entry in the t-table with the word pair constraint. We simply modify the GIZA++ toolkit (Och and Ney, 2003) by always weighting lexicon probabilities with soft constraints during iterative model training, and obtain 0.7% TER reduction on both sets and 0.4% BLEU improvement on the test set.

3.3 Analysis

To understand how prior knowledge encoded as soft constraints plays a role in guiding word alignment training, we compare statistics of different word alignment models. We find that our baseline HMM

Table 1: Translation Results with different word alignments.

Alignments	BLEU		TER	
	dev	test	dev	test
Model-4	0.310	0.296	0.528	0.530
+Mix	0.306	0.300	0.521	0.523
HMM	0.289	0.288	0.543	0.542
+Entropy	0.289	0.293	0.534	0.536
+BiLSA-1	0.294	0.300	0.531	0.521
+BiLSA-2	0.298	0.292	0.530	0.528
+Mix	0.302	0.304	0.532	0.524

generates 2.6% less number of total word links than that of Model-4. Part of the reason is that models of two directions in the baseline are trained simultaneously. The requirement of bi-directional evidence places a certain constraint on word alignments. When "BiLSA-1" constraints are applied in the baseline model, 2.7% less number of total word links are hypothesized, and consequently, less number of Arabic n-gram translations in the final phrase translation table are induced. The observation suggests that the constraints improve word alignment precision and accuracy of phrase translation tables as well.

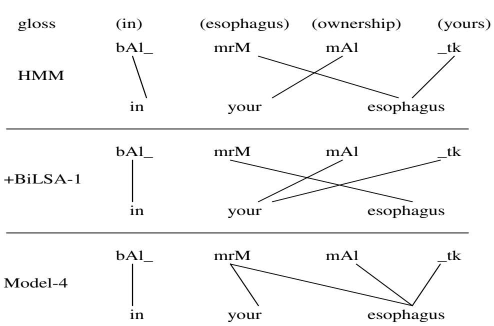

Figure 4: An example of word alignments under different models

Figure 4 shows example word alignments of a partial sentence pair. The complete English sentence is "have you ever had like any reflux diseases in your esophagus". We notice that the Arabic word "mrM" (means esophagus) appears only once in the corpus. Some of the word pair constraints are listed in Table 2. The example demos that due to reasonable constraints placed in word alignment training, the link to "_tK" is corrected and consequently we have accurate word translation for the Arabic singleton "mrM".

Table 2: Word pair constraint values

English e	Arabic f	$Con_{BiLSA-1}(f, e)$
esophagus	mrM	0.6424
	mAl	0.1819
	_tk	0.2897
your	mrM	0.6319
	mAl	0.4930
	_tk	0.9672

4 Related Work

Heuristics based on co-occurrence analysis, such as point-wise mutual information or Dice coefficients , have been shown to be indicative for word alignments (Zhang and Vogel, 2005; Melamed, 2000). The framework presented in this paper demonstrates the possibility of taking heuristics as constraints guiding statistical generative word alignment model training. Their effectiveness can be expected especially when data sparseness is severe.

Discriminative word alignment models, such as Ittycheriah and Roukos (2005); Moore (2005); Blunsom and Cohn (2006), have received great amount of study recently. They have proven that linguistic knowledge is useful in modeling word alignments under log-linear distributions as morphological, semantic or syntactic features. Our framework proposes to exploit these features differently by taking them as soft constraints of translation lexicon under a generative model.

While word alignments can help identifying semantic relations (van der Plas and Tiedemann, 2006), we proceed in the reverse direction. We investigate the impact of semantic constraints on statistical word alignment models as prior knowledge. In (Ma et al., 2004), bilingual semantic maps are constructed to guide word alignment. The framework we proposed seamlessly integrates derived semantic similarities into a statistical word alignment model. And we extended monolingual latent semantic analysis in bilingual applications.

Toutanova et al. (2002) augmented bilingual sentence pairs with part-of-speech tags as linguistic constraints for HMM-based word alignments. The constraints between tags are automatically learned in a parallel generative procedure along with lexicon. We have introduced hidden tags between a word pair to specialize their soft constraints, which serve as prior knowledge that will be used in guiding word alignment model training. Constraint between tags are embedded into the word to word generative process.

5 Conclusions and Future Work

We have presented a simple and effective framework to incorporate prior knowledge such as heuristics or linguistic features into statistical generative word alignment models. Prior knowledge serves as soft constraints that shall be placed on translation lexicon to guide word alignment model training and disambiguation during Viterbi alignment process. We studied two types of constraints that can be obtained automatically from data and showed improved performance (up to 1.6% absolute BLEU increase or 2.1% absolute TER reduction) in translating dialectical Arabic into English. Future work includes implementing the idea in alternative alignment models and also exploiting prior knowledge derived from such as manually-aligned data and pre-existing linguistic resources.

Acknowledgement We thank Mohamed Afify for discussions and the anonymous reviewers for suggestions.

References

J. R. Bellegarda. 2000. Exploiting latent semantic information in statistical language modeling. *Proc. of the IEEE*, 88(8):1279–1296, August.

M. Berry, T. Do, and S. Varadhan. 1993. Svdpackc (version 1.0) user's guide. Tech. report cs-93-194, University of Tennessee, Knoxville, TN.

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proc. of COLING/ACL*, pages 65–72.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312.

S. F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL*, pages 310–318.

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

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proc. of COLING/ACL*, pages 769–776.

A. Ittycheriah and S. Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proc. of HLT/EMNLP*, pages 89–96.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*.

LDC, 2002. *Buckwalter Arabic Morphological Analyzer Version 1.0*. LDC Catalog Number LDC2002L49.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Proc. of HLT/NAACL*, pages 104–111.

Q. Ma, K. Kanzaki, Y. Zhang, M. Murata, and H. Isahara. 2004. Self-organizing semantic maps and its application to word alignment in japanese-chinese parallel corpora. *Neural Netw.*, 17(8-9):1241–1253.

I. Dan. Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

R. C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proc. of HLT/EMNLP*, pages 81–88.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*, pages 295–302.

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

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*.

K. Toutanova, H. T. Ilhan, and C. Manning. 2002. Extentions to HMM-based statistical word alignment models. In *Proc. of EMNLP*.

Lonneke van der Plas and Jörg Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proc. of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 866–873.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM based word alignment in statistical translation. In *Proc. of COLING*.

B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, and J. Makhoul. 2006. Morphological decomposition for arabic broadcast news transcription. In *Proc. of ICASSP*, pages 1089–1092.

R. Zens, E. Matusov, and H. Ney. 2004. Improved word alignment using a symmetric lexicon model. In *Proc. of COLING*, pages 36–42.

Y. Zhang and S. Vogel. 2005. Competitive grouping in integrated phrase segmentation and alignment model. In *Proc. of the ACL Workshop on Building and Using Parallel Texts*, pages 159–162.