Deep Grammars in a Tree Labeling Approach to Syntax-based Statistical Machine Translation

Mark Hopkins
Department of Linguistics
University of Potsdam, Germany
hopkins@ling.uni-potsdam.de

Jonas Kuhn
Department of Linguistics
University of Potsdam, Germany
kuhn@ling.uni-potsdam.de

Abstract

In this paper, we propose a new syntax-based machine translation (MT) approach based on reducing the MT task to a tree-labeling task, which is further decomposed into a sequence of simple decisions for which discriminative classifiers can be trained. The approach is very flexible and we believe that it is particularly well-suited for exploiting the linguistic knowledge encoded in deep grammars whenever possible, while at the same time taking advantage of data-based techniques that have proven a powerful basis for MT, as recent advances in statistical MT show.

A full system using the Lexical-Functional Grammar (LFG) parsing system XLE and the grammars from the Parallel Grammar development project (ParGram; (Butt et al., 2002)) has been implemented, and we present preliminary results on English-to-German translation with a tree-labeling system trained on a small subsection of the Europarl corpus.

1 Motivation

Machine translation (MT) is probably the oldest application of what we call deep linguistic processing techniques today. But from its inception, there have been alternative considerations of approaching the task with data-based statistical techniques (cf. Warren Weaver's well-known memo from 1949). Only with fairly recent advances in computer technology have researchers been able to build effective statistical MT prototypes, but in the last few years, the statistical approach has received enormous research interest and made significant progress.

The most successful statistical MT paradigm has been, for a while now, the so-call phrase-based MT approach (Och and Ney, 2003). In this paradigm, sentences are translated from a source language to a target language through the repeated substitution of contiguous word sequences ("phrases") from the source language for word sequences in the target language. Training of the phrase translation model builds on top of a standard statistical word alignment over the training corpus of parallel text (Brown et al., 1993) for identifying corresponding word blocks, assuming no further linguistic analysis of the source or target language. In decoding, i.e. the application of the acquired translation model to unseen source sentences, these systems then typically rely on n-gram language models and simple statistical reordering models to shuffle the phrases into an order that is coherent in the target language.

An obvious advantage of statistical MT approaches is that they can adopt (often very idiomatic) translations of mid- to high-frequency constructions without requiring any language-pair specific engineering work. At the same time it is clear that a linguistics-free approach is limited in what it can ultimately achieve: only linguistically informed systems can detect certain generalizations from lower-frequency constructions in the data and successfully apply them in a similar but different linguistic context. Hence, the idea of "hybrid" MT, exploiting both linguistic and statistical information is fairly old. Here we will not consider classical, rule-based systems with some added data-based resource acquisition (although they may be among the best candidates for high-quality special-purpose translation – but adaption to new language pairs and sub-languages is very costly for these systems). The other form of hybridization – a statistical MT model that is based on a deeper analysis of the syntactic

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 33–40,
Ann Arbor, June 2005.

structure of a sentence – has also long been identified as a desirable objective in principle (consider (Wu, 1997; Yamada and Knight, 2001)). However, attempts to retrofit syntactic information into the phrase-based paradigm have not met with enormous success (Koehn et al., 2003; Och et al., 2003)[1], and purely phrase-based MT systems continue to outperform these syntax/phrase-based hybrids.

In this work, we try to make a fresh start with syntax-based statistical MT, discarding the phrase-based paradigm and designing a MT system from the ground up, using syntax as our central guiding star – besides the word alignment over a parallel corpus. Our approach is compatible with and can benefit substantially from rich linguistic representations obtained from deep grammars like the ParGram LFGs. Nevertheless, contrary to classical interlingual or deep transfer-based systems, the generative stochastic model that drives our system is grounded only in the cross-language word alignment and a surface-based phrase structure tree for the source language and will thus degrade gracefully on input with parsing issues – which we suspect is an important feature for making the overall system competitive with the highly general phrase-based MT approach.

Preliminary evaluation of our nascent system indicates that this new approach might well have the potential to finally realize some of the promises of syntax in statistical MT.

2 General Task

We want to build a system that can learn to translate sentences from a source language to a destination language. The general set-up is simple.

Firstly, we have a training corpus of paired sentences f and e, where target sentence e is a gold standard translation of source sentence f. These sentence pairs are annotated with auxiliary information, which can include word alignments and syntactic information. We refer to these annotated sentence pairs as *complete translation objects*.

Secondly, we have an evaluation corpus of source sentences. These sentences are annotated with a subset of the auxiliary information used to annotate the training corpus. We refer to these partially annotated source sentences as *partial translation objects*.

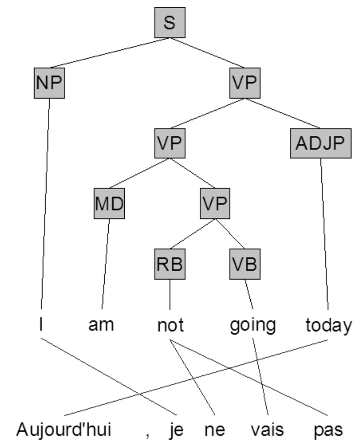

Figure 1: Example translation object.

The task at hand: use the training corpus to learn a procedure, through which we can successfully induce a complete translation object from a partial translation object. This is what we will define as *translation*.

3 Specific Task Addressed by this Paper

Before going on to actually describe a translation procedure (and how to induce it), we need to specify our prior assumptions about how the translation objects will be annotated. For this paper, we want to exploit the syntax information that we can gain from an LFG-parser, hence we will assume the following annotations:

(1) In the training and evaluation corpora, the source sentences will be parsed with the XLE-parser. The attribute-value information from LFG's f-structure is restructured so it is indexed by (c-structure) tree nodes; thus a tree node can bear multiple labels for various pieces of morphological, syntactic and semantic information.

(2) In the training corpus, the source and target sentence of every translation object will be aligned using GIZA++ (http://www.fjoch.com/).

In other words, our complete translation objects will be aligned tree-string pairs (for instance, Figure 1), while our partial translation objects will be trees (the tree portion of Figure 1). No other annotations will be assumed for this paper.

[1](Chiang, 2005) also reports that with his hierarchical generalization of the phrase-based approach, the addition of parser information doesn't lead to any improvements.

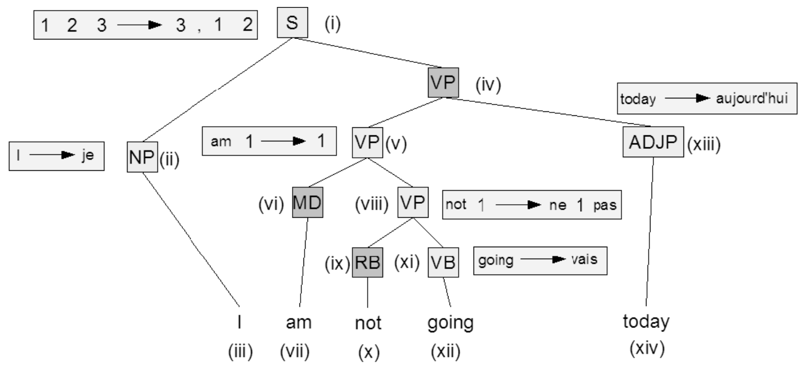

Figure 2: GHKM tree equivalent of example translation object. The light gray nodes are rule nodes of the GHKM tree.

4 Syntax MT as Tree Labeling

It is not immediately clear how one would learn a process to map a parsed source sentence into an aligned tree-string pair. To facilitate matters, we will map complete translation objects to an alternate representation. In (Galley et al., 2003), the authors give a semantics to aligned tree-string pairs by associating each with an annotated parse tree (hereafter called a *GHKM tree*) representing a specific theory about how the source sentence was translated into the destination sentence.

In Figure 1, we show an example translation object and in Figure 2, we show its associated GHKM tree. The GHKM tree is simply the parse tree f of the translation object, annotated with rules (hereafter referred to as *GHKM rules*). We will not describe in depth the mapping process from translation object to GHKM tree. Suffice it to say that the alignment induces a set of intuitive translation rules. Essentially, a rule like: "not 1 → ne 1 pas" (see Figure 2) means: if we see the word "not" in English, followed by a phrase already translated into French, then translate the entire thing as the word "ne" + the translated phrase + the word "pas." A parse tree node gets labeled with one of these rules if, roughly speaking, its span is still contiguous when projected (via the alignment) into the target language.

The advantage of using the GHKM interpretation of a complete translation object is that our translation task becomes simpler. Now, to induce a complete translation object from a partial translation object (parse tree), all we need to do is label the nodes of the tree with appropriate rules. We have reduced the vaguely defined task of translation to the concrete task of tree labeling.

5 The Generative Process

At the most basic level, we could design a naive generative process that takes a parse tree and then makes a series of decisions, one for each node, about what rule (if any) that node should be assigned. However it is a bit unreasonable to expect to learn such a decision without breaking it down somewhat, as there are an enormous number of rules that could potentially be used to label any given parse tree node. So let's break this task down into simpler decisions. Ideally, we would like to devise a generative process consisting of decisions between a small number of possibilities (ideally binary decisions).

We will begin by deciding, for each node, whether or not it will be annotated with a rule. This is clearly a binary decision. Once a generative process has made this decision for each node, we get a convenient byproduct. As seen in Figure 3, the LHS of each rule is already determined. Hence after this sequence of binary decisions, half of our task is already completed.

The question remains: how do we determine the RHS of these rules? Again, we could create a generative process that makes these decisions directly, but choosing the RHS of a rule is still a rather wide-open decision, so we will break it down further. For each rule, we will begin by choosing the *template* of its RHS, which is a RHS in which all sequences of variables are replaced with an empty slot into which variables can later be placed. For instance, the tem-

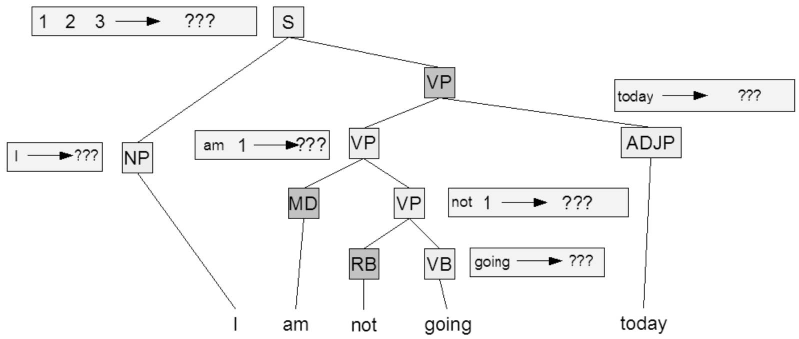

Figure 3: Partial GHKM tree, after rule nodes have been identified (light gray). Notice that once we identify the rule node, the rule left-hand sides are already determined.

plate of ⟨"ne", x_1, "pas"⟩ is ⟨"ne", X, "pas"⟩ and the template of ⟨x_3, ",", x_1, x_2⟩ is ⟨X, ",", X⟩, where X represents the empty slots.

Once the template is chosen, it simply needs to be filled with the variables from the LHS. To do so, we process the LHS variables, one by one. By default, they are placed to the right of the previously placed variable (the first variable is placed in the first slot). We repeatedly offer the option to push the variable to the right until the option is declined or it is no longer possible to push it further right. If the variable was not pushed right at all, we repeatedly offer the option to push the variable to the left until the option is declined or it is no longer possible to push it further left. Figure 4 shows this generative story in action for the rule RHS ⟨x_3, ",", x_1, x_2⟩.

These are all of the decisions we need to make in order to label a parse tree with GHKM rules. A trace of this generative process for the GHKM tree of Figure 2 is shown in Figure 5. Notice that, aside from the template decisions, all of the decisions are binary (i.e. feasible to learn discriminatively). Even the template decisions are not terribly large-domain, if we maintain a separate feature-conditional distribution for each LHS template. For instance, if the LHS template is ⟨"not", X⟩, then RHS template ⟨"ne", X, "pas"⟩ and a few other select candidates should bear most of the probability mass.

Decision to make	Decision	RHS so far
RHS template?	X , X	X , X
default placement of var 1		1 , X
push var 1 right?	yes	X , 1
default placement of var 2		X , 1 2
push var 2 left?	no	X , 1 2
default placement of var 3		X , 1 2 3
push var 3 left?	yes	X , 1 3 2
push var 3 left?	yes	X , 3 1 2
push var 3 left?	yes	3 , 1 2

Figure 4: Trace of the generative story for the right-hand side of a GHKM rule.

5.1 Training

Having established this generative story, training is straightforward. As a first step, we can convert each complete translation object of our training corpus to the trace of its generative story (as in Figure 5). These decisions can be annotated with whatever feature information we might deem helpful. Then we simply divide up these feature vectors by decision type (for instance, rule node decisions, template decisions, etc.) and train a separate discriminative classifier for each decision type from the feature vectors. This method is quite flexible, in that it allows us to use any generic off-the-shelf classification software to train our system. We prefer learners that produce distributions (rather than hard classifiers) as output, but this is not required.

5.2 Exploiting deep linguistic information

The use of discriminative classifiers makes our approach very flexible in terms of the information that can be exploited in the labeling (or translation) process. Any information that can be encoded as features relative to GHKM tree nodes can be used. For the experiments reported in this paper, we parsed the source language side of a parallel corpus (a small subsection of the English-German Europarl corpus; (Koehn, 2002)) with the XLE system, using the ParGram LFG grammar and applying probabilistic disambiguation (Riezler et al., 2002) to obtain a single analysis (i.e., a c-structure [phrase structure tree] and an f-structure [an associated attribute-value matrix with morphosyntactic feature information and a shallow semantic interpretation]) for each sentence. A fall-back mechanism integrated in the parser/grammar ensures that even for sentences that do not receive a full parse, substrings are deeply parsed and can often be treated successfully.

We convert the c-structure/f-structure representation that is based on XLE's sophisticated word-internal analysis into a plain phrase structure tree representation based on the original tokens in the source language string. The morphosyntactic feature information from f-structure is copied as additional labeling information to the relevant GHKM tree nodes, and the f-structural dependency relation among linguistic units is translated into a relation among corresponding GHKM tree nodes. The relational information is then used to systematically extend the learning feature set for the tree-node based classifiers.

In future experiments, we also plan to exploit linguistic knowledge about the target language by factorizing the generation of target language words into separate generation of lemmas and the various morphosyntactic features. In decoding, a morphological generator will be used to generate a string of surface words.

5.3 Decoding

Because we have purposely refused to make any Markov assumptions in our model, decoding cannot be accomplished in polynomial time. Our hypothesis is that it is better to find a suboptimal solution of a high-quality model than the optimal solution of a poorer model. We decode through a simple search through the space of assignments to our generative process.

This is, potentially, a very large and intractible search space. However, if most assignment decisions can be made with relative confidence (i.e. the classifiers we have trained make fairly certain decisions), then the great majority of search nodes have values which are inferior to those of the best solutions. The standard search technique of *depth-first branch-and-bound search* takes advantage of search spaces with this particular characteristic by first finding greedy good-quality solutions and using their values to optimally prune a significant portion of the search space. Depth-first branch-and-bound search has the following advantage: it finds a good (suboptimal) solution in linear time and continually improves on this solution until it finds the optimal. Thus it can be run either as an optimal decoder or as a heuristic decoder, since we can interrupt its execution at any time to get the best solution found so far. Additionally, it takes only linear space to run.

6 Preliminary results

In this section, we present some preliminary results for an English-to-German translation system based on the ideas outlined in this paper.

Our data was a subset of the Europarl corpus consisting of sentences of lengths ranging from 8 to 17 words. Our training corpus contained 50000 sentences and our test corpus contained 300 sentences. We also had a small number of reserved sentences for development. The English sentences were parsed with XLE, using the English ParGram LFG grammar, and the sentences were word-aligned with GIZA++. We used the WEKA machine learning package (Witten and Frank, 2005) to train the distributions (specifically, we used model trees).

For comparison, we also trained and evaluated the phrase-based MT system Pharaoh (Koehn, 2005) on this limited corpus, using Pharaoh's default parameters. In a different set of MT-as-Tree-Labeling experiments, we used a standard treebank parser trained on the PennTreebank Wall Street Journal section. Even with this parser, which produces less detailed information than XLE, the results are competitive when assessed with quantitative measures: Pharaoh achieved a BLEU score of 11.17 on the test set, whereas our system achieved a BLEU score of 11.52. What is notable here is not the scores themselves (low due to the size of the training corpus). However our system managed to perform comparably with Pharaoh in a very early stage of its development, with rudimentary features and without the benefit of an n-gram language model.

For the XLE-based system we cannot include quantitative results for the same experimental setup at the present time. As a preliminary qualitative

Decision to make	Decision	Active features
rule node (i)?	YES	NT="S"; HEAD = "am"
rule node (ii)?	YES	NT="NP"; HEAD = "I"
rule node (iv)?	NO	NT="VP"; HEAD = "am"
rule node (v)?	YES	NT="VP"; HEAD = "am"
rule node (vi)?	NO	NT="MD"; HEAD = "am"
rule node (viii)?	YES	NT="VP"; HEAD = "going"
rule node (ix)?	NO	NT="RB"; HEAD = "not"
rule node (xi)?	YES	NT="VB"; HEAD = "going"
rule node (xiii)?	YES	NT="ADJP"; HEAD = "today"
RHS template? (i)	X , X	NT="S"
push var 1 right? (i)	YES	VARNT="NP"; PUSHPAST= ","
push var 2 left? (i)	NO	VARNT="VP"; PUSHPAST= "NP"
push var 3 left? (i)	YES	VARNT="ADJP"; PUSHPAST= "VP"
push var 3 left? (i)	YES	VARNT="ADJP"; PUSHPAST= "NP"
push var 3 left? (i)	YES	VARNT="ADJP"; PUSHPAST= ","
RHS template? (ii)	je	NT="NP"; WD="I"
RHS template? (v)	X	NT="VP"
RHS template? (viii)	ne X pas	NT="VP"; WD="not"
RHS template? (xi)	vais	NT="VB"; WD="going"
RHS template? (xiii)	aujourd'hui	NT="ADJP"; WD="today"

Figure 5: Trace of a top-down generative story for the GHKM tree in Figure 2.

evaluation, let's take a closer look at the sentences produced by our system, to gain some insight as to its current strengths and weaknesses.

Starting with the English sentence (1) (note that all data is lowercase), our system produces (2).

(1) i agree with the spirit of those amendments .

(2) ich stimme die geist dieser änderungsanträge
I vote the.FEM spirit.MASC these change-proposals
zu .
to .

The GHKM tree is depicted in Figure 6. The key feature of this translation is how the English phrase "agree with" is translated as the German "stimme ... zu" construction. Such a feat is difficult to produce consistently with a purely phrase-based system, as phrases of arbitrary length can be placed between the words "stimme" and "zu", as we can see happening in this particular example. By contrast, Pharaoh opts for the following (somewhat less desirable) translation:

(3) ich stimme mit dem geist dieser
I vote with the.MASC spirit.MASC these
änderungsanträge .
change-proposals .

A weakness in our system is also evident here. The German noun "Geist" is masculine, thus our system uses the wrong article (a problem that Pharaoh, with its embedded n-gram language model, does not encounter).

In general, it seems that our system is superior to Pharaoh at figuring out the proper way to arrange the words of the output sentence, and inferior to Pharaoh at finding what the actual translation of those words should be.

Consider the English sentence (4). Here we have an example of a modal verb with an embedded infinitival VP. In German, infinite verbs should go at the end of the sentence, and this is achieved by our system (translating "shall" as "werden", and "submit" as "vorlegen"), as is seen in (5).

(4)) we shall submit a proposal along these lines before the end of this year .

(5) wir werden eine vorschlag in dieser
we will a.FEM proposal.MASC in these
haushaltslinien vor die ende dieser
budget-lines before the.FEM end.NEUT this.FEM
jahres vorlegen .
year.NEUT submit .

Pharaoh does not manage this (translating "submit" as "unterbreiten" and placing it mid-sentence).

(6) werden wir unterbreiten eine vorschlag in dieser
will we submit a proposal in these
haushaltslinien vor ende dieser jahr .
budget-lines before end this.FEM year.NEUT .

It is worth noting that while our system gets the word order of the output system right, it makes sev-

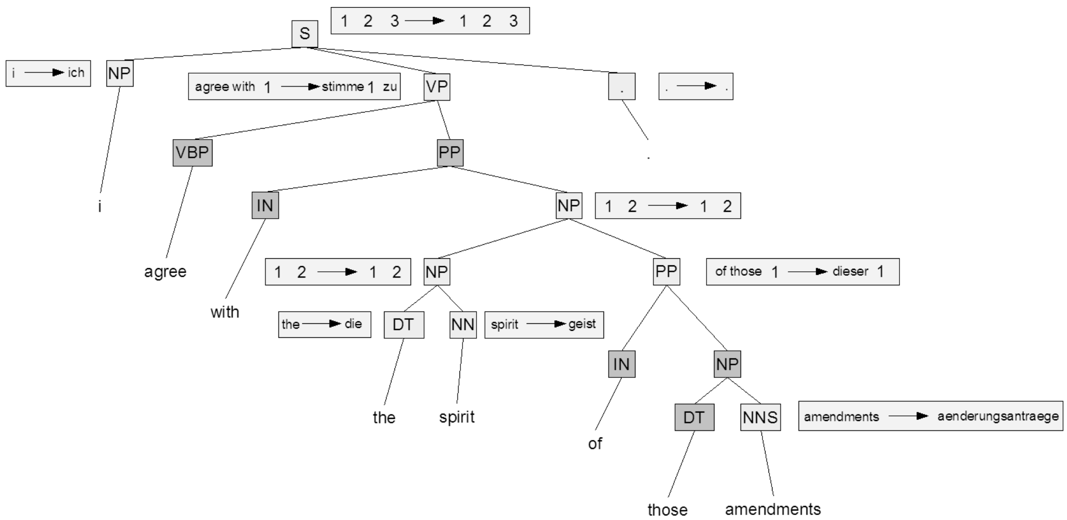

Figure 6: GHKM tree output for a test sentence.

eral agreement mistakes and (like Pharaoh) doesn't get the translation of "along these lines" right.

In Figure 7, we show sample translations by the three systems under discussion for the first five sentences in our evaluation set. For the LFG-based approach, we can at this point present only results for a version trained on 10% of the sentence pairs. This explains why more source words are left untranslated. But note that despite the small training set, the word ordering results are clearly superior for this system: the syntax-driven rules place the untranslated English words in the correct position in terms of German syntax.

The translations with Pharaoh contain relatively few agreement mistakes (note that it exploits a language model of German trained on a much larger corpus). The phrase-based approach does however skip words and make positioning mistakes some of which are so serious (like in the last sentence) that they make the result hard to understand.

7 Discussion

In describing this pilot project, we have attempted to give a "big picture" view of the essential ideas behind our system. To avoid obscuring the presentation, we have avoided many of the implementation details, in particular our choice of features. There are exactly four types of decisions that we need to train: (1) whether a parse tree node should be a rule node, (2) the RHS template of a rule, (3) whether a rule variable should be pushed left, and (4) whether a rule variable should be pushed right. For each of these decisions, there are a number of possible features that suggest themselves. For instance, recall that in German, embedded infinitival verbs get placed at the end of the sentence or clause. So when the system is considering whether to push a rule's noun phrase to the left, past an existing verb, it would be useful for it to consider (as a feature) whether that verb is the first or second verb of its clause and what the morphological form of the verb is.

Even in these early stages of development, the MT-as-Tree-Labeling system shows promise in using syntactic information flexibly and effectively for MT. Our preliminary comparison indicates that using deep syntactic analysis leads to improved translation behavior. We hope to develop the system into a competitive alternative to phrase-based approaches.

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.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel gram-

source	we believe that this is a fundamental element .
professional translation	wir denken , dass dies ein grundlegender aspekt ist .
PHARAOH (50k)	wir halten dies für eine grundlegende element .
TL-WSJ (50k)	wir glauben , dass diesen ist ein grundlegendes element .
TL-LFG (5k)	wir meinen , dass dies eine grundlegende element ist .
source	it is true that lisbon is a programme for ten years .
professional translation	nun ist lissabon ein programm für zehn jahre .
PHARAOH (50k)	es ist richtig , dass lissabon ist eine programm für zehn jahren .
TL-WSJ (50k)	es ist richtig , dass lissabon ist eine programm für zehn jahren .
TL-LFG (5k)	es ist *true* , dass *lisbon* eine programm für zehn jahren ist .
source	i completely agree with each of these points .
professional translation	ich bin mit jeder einzelnen dieser aussagen voll und ganz einverstanden .
PHARAOH (50k)	ich ..... völlig einverstanden mit jedem dieser punkte .
TL-WSJ (50k)	ich bin völlig mit jedes diese fragen einer meinung .
TL-LFG (5k)	ich *agree completely* mit jeder dieser punkte .
source	however , i would like to add one point .
professional translation	aber ich möchte gern einen punkt hinzufügen .
PHARAOH (50k)	allerdings möchte ich noch eines sagen .
TL-WSJ (50k)	ich möchte jedoch ~~an~~ noch einen punkt hinzufügen .
TL-LFG (5k)	allerdings möchte ich einen punkt *add* .
source	this is undoubtedly a point which warrants attention .
professional translation	ohne jeden zweifel ist dies ein punkt , der aufmerksamkeit verdient .
PHARAOH (50k)	das ist sicherlich eine punkt .... rechtfertigt das aufmerksamkeit .
TL-WSJ (50k)	das ist ohne zweifel eine punkt , die *warrants* beachtung .
TL-LFG (5k)	das ist *undoubtedly* .... sache , die *attention warrants* .

Figure 7: Sample translations by (1) the PHARAOH system, (2) our system with a treebank parser (TL-WSJ), (3) our system with the XLE parser (TL-LFG). (1) and (2) were trained on 50,000 sentence pairs, (3) just on (3) sentence pairs. Error coding: wrong morphological form, incorrectly positioned word, *untranslated source word*, missed word: ...., ~~extra word~~.

mar project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2003. What's in a translation rule? In *Proc. NAACL*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada.

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

Philipp Koehn. 2005. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

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

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, Viren Jain, Z.Jin, and D. Radev. 2003. Syntax for statistical machine translation. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore. Summer Workshop Final Report.

Stefan Riezler, Dick Crouch, Ron Kaplan, Tracy King, John Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Pennsylvania, Philadelphia*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

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

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530.