Syntax-aware Phrase-based Statistical Machine Translation: System Description

Ulrich Germann
University of Toronto
Toronto, Ontario, Canada
germann@cs.toronto.edu

Abstract

We present a variant of phrase-based SMT that uses source-side parsing and a constituent reordering model based on word alignments in the word-aligned training corpus to predict hierarchical block-wise reordering of the input. Multiple possible translation orders are represented compactly in a source order lattice. This source order lattice is then annotated with phrase-level translations to form a lattice of tokens in the target language. Various feature functions are combined in a log-linear fashion to evaluate paths through that lattice.

1 Introduction

Dealing with word order differences is one of the major challenges in automatic translation between human languages. With its moderate context sensitivity and reliance on n-gram language models, *phrase-based statistical machine translation* (PBSMT) (Koehn *et al.*, 2003) is usually quite good at performing small word order changes — for instance, the inversion of adjective and noun in English-to-French translation and vice versa. However, it regularly fails to execute word order changes over long distances, as they are required, for example, to accommodate the substantial differences in the word order in subordinate clauses between German and English, or to cope with the phenomenon of the “sentence bracket” (*Satzklammer*) in German main clauses, in which the finite part of the verb complex and additional elements (separable prefixes, participles, infinitives, etc.) form a bracket that encloses most of the arguments and other adverbial constituents, as shown in Fig. 1. In order to keep decoding complexity in check, phrase-based decoders such as the *Moses* system (Koehn *et al.*, 2007) routinely limit the maximum distance for word order changes to six or seven word positions, thus ruling out, a priori, word order changes necessary to achieve good and fluent translations.

As is generally acknowledged, word order differences are not entirely arbitrary. By and large they follow syntactic structure. An analysis of word-aligned French-English data by Fox (2002) showed that word alignment links rarely cross syntactic boundaries. Wu’s (1997) *Inversion Transaction Grammar* (ITG), assumes that word order differences can be accounted for by hierarchical inversion of adjacent blocks of text. Yamada and Knight (2001) present a stochastic model for transforming English parse trees into Japanese word sequences within a source-channel framework for Japanese-to-English translation. Collins *et al.* (2005) perform heuristic word re-ordering from German into English word order based on German parse trees with a particular focus on the aforementioned drastic word order differences between German and English clause structure.

Building on Chiang (2007), several systems under active development (e.g., Weese *et al.*, 2011; Dyer *et al.*, 2010) rely on synchronous context-free grammars to deal with word order differences. In essence, these systems parse the input while synchronously building a parse tree in the translation target language, using probabilities of the source and target trees as well as correspondence probabilities to evaluate translation hypotheses.

"Dieser$_1$ Vorschlag$_2$ $\boxed{\text{wird}_3}$ sicherlich$_4$ im$_5$ Ausschuß$_6$ gründlich$_7$ $\boxed{\text{diskutiert}_8 \text{ werden}_9 \text{ müssen}_{10}}$."

"This$_1$ proposal$_2$ will$_3$ certainly$_4$ have$_{10}$ to$_{10}$ be$_9$ discussed$_8$ toroughly$_7$ in$_5$ the$_5$ commission$_6$."

Figure 1: The sentence bracket (*Satzklammer*) in German.

The system presented in this paper takes a slightly different route and is closer to the approach taken by Collins *et al.* (2005): we parse only monolingually on the source side, re-order, and then translate. However, unlike Collins *et al.* we do not use a series of rules to perform the transformations (nor do we re-order the training data on the source side), but try to learn reordering rules from the word-aligned corpus with the original word order on both sides. Moreover, we do not commit to a single parse and a single re-ordering of the source at translation time but consider multiple parse alternatives to create a lattice of possible translation orders. Each vertex in the lattice corresponds to a specific subset of source words translated up to that point.

Individual edges and sequences of edges in this lattice are annotated with word- and phrase[1]-level translations extracted from the word-aligned training corpus, in the same way as phrase tables for PBSMT are constructed[2]. An optimal path through the lattice is determined by dynamic programming, considering a variety of feature functions combined in a log-linear fashion.

In the following, we first describe the individual processing steps in more detail and then try to shed some light on the system's performance in this year's shared task. Due to space limitations, many details will have to be skipped.

2 System Description

2.1 Grammatical framework

The central idea underlying this work is that grammar constrains word reordering: we are allowed to permute siblings in a CFG tree, or the governor and its dependents in a dependency structure, but we are not allowed to break phrase coherence by moving words out of their respective sub-tree. Obviously, we need to be careful in the precise formulation of our grammar, so as not to over-constrain word order options. For example, the German parse tree for the phrase *ein*$_1$ [*zu hoher*]$_2$ *Preis*$_3$ in Fig. 2 below rules out the proper word order of its English translation [*too high*]$_2$ *a*$_1$ *price*$_3$.

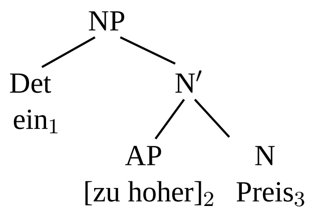

Figure 2: X-bar syntax can be too restrictive. This tree does not allow the word order of the English translation [*too high*]$_2$ *a*$_1$ *price*$_3$.

In her analysis of phrasal cohesion in translation, Fox (2002) pointed out that phrasal cohesion is greater with respect to dependency structure than with respect to constituent structure. We therefore decided to rely on the segmentation granularity inherent in dependency parses.

2.2 Parsing

For parsing, we developed our own hybrid left-corner dependency parser for German. In many respects, it is inspired by the work on dependency parsing by Eisner (1996) (edge factorization) and McDonald *et al.* (2005) (choice of features for edge scores). From the generative point of view, we can imagine the following generative process: We start with the root word of the sentence. A Markov process then generates this word's immediate dependents from left to right, at some point placing the head word itself. The dependents (but not the head word) are then expanded recursively in the same fashion. At parse time we process the input left to right, deciding for each word what its governors are, or whether it governs some items to its left or right.

[1]"Phrase" being any contiguous sequence of words in this context, as in PBSMT.

[2]Except that we do not pre-compute phrase tables but construct them dynamically on the fly using suffix arrays, as suggested by Callison-Burch *et al.* (2005).

Since each word has exactly one governor (bar the root word), we renormalize edge scores by marginalizing over the potential governors. If the word is potentially the left corner of a sub-tree, we establish a new rule (akin to a dotted rule in an Earley parser) and add it to the parse chart. For potential governors to the left, we scan the parse chart for partial productions that end immediately before the word in question and extend them by the word in question. Whenever we add an item to a partial production that is "past or reaching its head" (i.e., the span covered by the rule includes the sub-tree's root or the newly added item is the root), we treat the sub-tree as a new item in a bottom-up fashion, i.e., determine potential governors outside of the span covered, add a new rule if the sub-tree could be the left corner of a larger sub-tree, etc. In addition to the joint probability of all individual edges, we also consider the cost of adding an item to a partial production. To reduce parse complexity, we use a beam to limit the number of potential governors that are considered for each item. Unlike conventional CFGs, the set of "rules" in this grammar is not finite; rules are generated on the fly by a Markov process. This adds robustness; we can always attach an item (token or sub-tree) to one of its immediate neighbors.

2.3 Construction of a source order lattice (SOL)

Rows and columns in the parse chart correspond to the start and end positions of parse spans in the sentence. Each cell contains zero or more production rules that correspond to different segmentations of the respective span into sub-spans that may be reordered during translation. Based on the underlying part-of-speech tags, we retrieve similar syntactic configurations from the word-aligned, source-side-parsed training corpus.

For each example retrieved from the training corpus, we determine, from the word alignment information in the training corpus, the order in which the dependents and the head word are translated. To reduce noise from alignment errors, each example is weighted by the joint lexical translation probability of the words immediately involved in the production (i.e., the head and its dependents, but not grandchildren). Thus, examples with unlikely word alignments count less than examples that have highly probable word alignments. If exact matches for the production rule in question cannot be found in the corpus (which happens frequently), we fall back on a factorized model that maps from source to target positions based on the part-of-speech of the dependent in question and its governor. Words that are part of the verb complex (auxiliaries, separable prefixes, the 'lexical head', etc.) are grouped together and receive special treatment. (This is currently work in progress; at this point, we translate only the lexical head, but ignore negation and auxiliaries.)

For each of the top N segmentations suggested by the parser, translation order probabilities are computed on the basis of the weighted occurrence counts, and used to set the edge weights in a lattice of possible translation orders, which we call the Source Order Lattice (SOL). Each vertex in this lattice corresponds to a specific set of source words translated so far. (In principle, the number of vertices in this lattice is exponential in the length of the input sentence; in practice, since we consider only a small number of possibilities, their number is quite manageable.) For each chunk of text in the suggested order of translation, we increase the weight of the edge between the vertex representing the set of words translated so far and the vertex representing the set of words translated after this chunk has been translated by the probability of translating the chunk in question at this particular point in the translation process. Edges representing two or more consecutive words (with the exception of those representing a verb complex) are recursively replaced by local SOLs, until each edge corresponds to a single word in the source sentence.

2.4 Constructing a target word lattice

The global SOL thus constructed is then transformed into a Target Word Lattice (TWL), while maintaining underlying alignment information. Each individual edge or sequence of adjacent edges corresponding to a contiguous sequence of words in the source sentence is replaced by a lattice that encodes the range of possible translations for the respective word or phrase. Translations are extracted from the word-aligned bilingual training corpus with the phrase-extraction method that is commonly used in phrase-based SMT. As it is done in the *Joshua* system (Weese *et al.*, 2011), we extract phrase translations on the fly from the word-aligned bilingual corpus using suffix arrays instead of using pre-computed phrase tables.

2.5 Search

Once constructed, the TWL is searched with dynamic programming with a beam search. Hypotheses are scored by a log-linear combination of the following feature functions. Feature values are normalized by hypothesis length unless noted otherwise, to safeguard against growth of cumulative feature values at different rates as the length of a hypothesis increases, and to keep hypotheses of different lengths mutually comparable.

- **Distortion probabilities** from the SOL as described above.
- **Relative phrase translation frequencies** based on counts in the training corpus.
- **Lexical translation probabilities**: forward ($\mathrm{p}\,(\textit{target}\,|\,\textit{source})$; normalized by target length) and backward ($\mathrm{p}\,(\textit{source}\,|\,\textit{target})$; normalized by source length). Lexical translation probabilities are based on alignment link counts in the word-aligned corpus.
- **N-gram language model probability** as estimated with the SRILM toolkit.
- **Fluency**. Simple length-based normalization of joint n-gram probabilities is problematic. It entices the decoder to "throw in" additional, highly frequent words to increase the language model score. Inversely, lack of normalization provides an incentive to keep translation hypotheses as short as possible, even at the expense of fluency. This fluency feature function computes the ratio of the language model probability of each proposed target word in context and its unigram probability. Rewards ($\mathrm{p}\,(w_i\,|\,w_{i-k+1}\ldots w_{i-1}) > \mathrm{p}\,(w_i)$) and penalties ($\mathrm{p}\,(w_i\,|\,w_{i-k+1}\ldots w_{i-1}) < \mathrm{p}\,(w_i)$) receive different weights in the log-linear combination. Rewards are normalized by target length; penalties by the number of source words translated. The rationale between the different forms of normalization is this: if we don't normalize rewards by hypothesis length, we have an incentive to pad the translation with highly frequent tokens (commas, 'the') wherever their probability in context is higher than their simple unigram probability. Awkwardly placed tokens, on the other hand, should always trigger a penalty, and the system should not be allowed to soften the blow by adding more poorly, but not quite as poorly placed tokens. Normalization of penalties by covered source length is an acknowledgement of the fact that in longer sentences, the probability of having points of disfluency increases. We use two reward/penalty pairs sets of fluency feature functions. One operates on surface forms, the other one on part-of-speech tag sequences.
- **Cumulative probability density of observed n-gram counts.** This feature function penalizes n-grams that do not occur as often as they should (even if observed), based on prior observation, and rewards those that do. Consider the following sequence of words in English:

  can you are

  The sequence *can you* is fairly frequent, and so is *you are*. However, *can you are* is not. With standard n-gram back-off models, the model, upon not finding the full context *can you* for *are*, will back off to the context *you* and thus assign an inappropriately high probability to $\mathrm{p}\,(\textit{are}\,|\,\textit{can you})$.

  The n-gram cdf feature models the event as a Bernoulli experiment. Suppose, for example, that $p\,(\textit{are}\,|\,\textit{you}) = .01$, and we have observed *can you* 1000 times, but have never seen *can you are*. Then the expected count of observations is 10 and

  $$\mathrm{cdf}\,(0\,|\,1000;.01) = (1-.01)^{1000} \approx .000043$$

3 Training and tuning

The system was trained on the German-English part of Europarl corpus (v.5). The language model for English was trained on all monolingual data available for WMT-2010. We true-cased, but did not lower-case the data. Word alignment was performed with multi-threaded Giza++ (Gao and Vogel, 2008).

In order to bootstrap training data for our parser, we parsed the German side of the Europarl corpus with the Berkeley Parser (Petrov *et al.*, 2006; Petrov and Klein, 2007) and converted the CFG structures to dependency structures using simple hand-written heuristics to identify the head in each phrase, similar to those used by Magerman (1995) and Collins (1996). This head was then selected as the governor of the respective phrase. Part-of-speech tagging and lemmatization on the English side as well as the German development and test data was performed with the tool TreeTagger (Schmid, 1995).

For tuning the model parameters, we tried to apply pairwise rank optimization (PRO) (Hopkins and May, 2011), but we were not able to achieve results that beat our hand-tuned parameter settings.

4 Evaluation

Unfortunately, with a BLEU score of .121, (.150 after several bug fixes in the program code), our system performed extremely poorly in the shared task. We have since tried to track down the reasons for the poor performance, but have not been able to find a compelling explanation for it.

A partial explanation may lie in the fact that we used only the Europarl data for training.[3] However, our system also lags far behind a baseline Moses system trained on the same subset of data used for our system, which achieves a BLEU score of .184.

Since our feature functions are very similar to those used in MOSES, we suspect that better tuning of the feature weights might close the gap. We are currently in the process of implementing and testing other parameter tuning methods (in addition to manual tuning and PRO), specifically lattice-based minimum error rate training (Macherey *et al.*, 2008) and batch MIRA (Cherry and Foster, 2012).

5 Conclusion

We have presented a variant of PBSMT that uses syntactic information from source-side parses in order to account better for word-order differences in German-to-English machine translation, while preserving the advantages of PBSMT. Several components were developed from scratch, such as a dependency parser for German and a reordering model for parse constituents, as well as several novel variants of n-gram based fluency measures. While our results for this year's shared task are certainly disappointing, we nevertheless believe that we are on the right track. We are not ready to give up quite yet.

[3]Participation in the shared task was a short term decision, and we did not have the time to re-train our system.

References

Callison-Burch, Chris, Colin Bannard, and Josh Schroeder. 2005. "Scaling phrase-based statistical machine translation to larger corpora and longer phrases." *43rd Annual Meeting of the Association for Computational Linguistics (ACL '05),* 255–262. Ann Arbor, Michigan.

Cherry, Colin and George Foster. 2012. "Batch tuning strategies for statistical machine translation." *2012 Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Quebéc, Canada.

Chiang, David. 2007. "Hierarchical phrase-based translation." *Computational Linguistics*, 33(2):1–28.

Collins, Michael, Philipp Koehn, and Kučerová Ivona. 2005. "Clause restructuring for statistical machine translation." *43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*. Ann Arbor, MI, USA.

Collins, Michael John. 1996. "A new statistical parser based on bigram lexical dependencies." *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics,* 184–191. Santa Cruz, California, USA.

Dyer, Chris, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models." *Proceedings of the ACL 2010 System Demonstrations*, 7–12. Uppsala, Sweden.

Eisner, Jason M. 1996. "Three new probabilistic models for dependency parsing: An exploration." *The 16th International Conference on Computational Linguistics (COLING '96),* 340–345. Copenhagen, Denmark.

Fox, Heidi J. 2002. "Phrasal cohesion and statistical machine translation." *Conference on Em-*

pirical Methods in Natural Language Processing (EMNLP '02), 304–311. Philadelphia, PA.

Gao, Qin and Stephan Vogel. 2008. "Parallel implementations of word alignment tool." *Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 49–57. Columbus, Ohio.

Hopkins, Mark and Jonathan May. 2011. "Tuning as ranking." *Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Edinburgh, UK.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. "Moses: Open source toolkit for statistical machine translation." *45th Annual Meeting of the Association for Computational Linguistics (ACL '07): Demonstration Session*. Prague, Czech Republic.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. "Statistical phrase-based translation." *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '03)*, 48–54. Edmonton, AB, Canada.

Macherey, Wolfgang, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. "Lattice-based minimum error rate training for statistical machine translation." *Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, 725–734. Honolulu, Hawaii.

Magerman, David M. 1995. "Statistical decision-tree models for parsing." *Proceedings of the Annual Meeting of the ACL*, 276–283.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. "Online large-margin training of dependency parsers." *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 91–98. Ann Arbor, Michigan.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. "Learning accurate, compact, and interpretable tree annotation." *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 433–440. Sydney, Australia.

Petrov, Slav and Dan Klein. 2007. "Improved inference for unlexicalized parsing." *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 404–411. Rochester, New York.

Schmid, Helmut. 1995. "Improvements in part-of-speech tagging with an application to German." *In Proceedings of the ACL SIGDAT-Workshop*, 47–50. Dublin, Ireland.

Weese, Jonathan, Juri Ganitkevitch, Chris Callison-Burch, Matt Post, and Adam Lopez. 2011. "Joshua 3.0: Syntax-based machine translation with the Thrax grammar extractor." *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 478–484. Edinburgh, Scotland.

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

Yamada, Kenji and Kevin Knight. 2001. "A syntax-based statistical translation model." *39th Annual Meeting of the Association for Computational Linguistics (ACL '01)*. Toulouse, France.