Linear Inversion Transduction Grammar Alignments as a Second Translation Path

Markus SAERS and **Joakim NIVRE**
Computational Linguistics Group
Dept. of Linguistics and Philology
Uppsala University
Sweden
first.last@lingfil.uu.se

Dekai WU
Human Language Technology Center
Dept. of Computer Science and Engineering
HKUST
Hong Kong
dekai@cs.ust.hk

Abstract

We explore the possibility of using Stochastic Bracketing Linear Inversion Transduction Grammars for a full-scale German–English translation task, both on their own and in conjunction with alignments induced with GIZA++. The rationale for transduction grammars, the details of the system and some results are presented.

1 Introduction

Lately, there has been some interest in using Inversion Transduction Grammars (ITGs) for alignment purposes. The main problem with ITGs is the time complexity, $\mathcal{O}(Gn^6)$ doesn't scale well. By limiting the grammar to a bracketing ITG (BITG), the grammar constant (G) can be eliminated, but $\mathcal{O}(n^6)$ is still prohibitive for large data sets.

There has been some work on approximate inference of ITGs. Zhang et al. (2008) present a method for evaluating spans in the sentence pair to determine whether they should be excluded or not. The algorithm has a best case time complexity of $\mathcal{O}(n^3)$. Saers, Nivre & Wu (2009) introduce a beam pruning scheme, which reduces time complexity to $\mathcal{O}(bn^3)$. They also show that severe pruning is possible without significant deterioration in alignment quality (as measured by downstream translation quality). Haghighi et al. (2009) use a simpler aligner as guidance for pruning, which reduces the time complexity by two orders of magnitude. Their work also partially implements the phrasal ITGs for translation-driven segmentation introduced in Wu (1997), although they only allow for one-to-many alignments, rather than many-to-many alignments. A more extreme approach is taken in Saers, Nivre & Wu (2010). Not only is the search severely pruned, but the grammar itself is limited to a linearized form, getting rid of branching within a single parse. Although a small deterioration in downstream translation quality is noted (compared to harshly pruned SBITGs), the grammar can be induced in linear time.

In this paper we apply SBLITGs to a full size German–English WMT'10 translation task. We also use differentiated translation paths to combine SBLITG translation models with a standard GIZA++ translation model.

2 Background

A transduction grammar is a grammar that generates a pair of languages. In a transduction grammar, the terminal symbols consist of pairs of tokens where the first is taken from the vocabulary of one of the languages, and the second from the vocabulary of the other. Transduction grammars have to our knowledge been restricted to transduce between languages no more complex than context-free languages (CFLs). Transduction between CFLs was first described in Lewis & Stearns (1968), and then further explored in Aho & Ullman (1972). The main motivation for exploring this was to build programming language compilers, which essentially translate between source code and machine code. There are two types of transduction grammars between CFLs described in the computer science literature: simple transduction grammars (STGs) and syntax-directed transduction grammars (SDTGs). The difference between them is that STGs are monotone, whereas SDTGs allow unlimited reordering in rule productions. Both allow the use of singletons to insert and delete tokens from either language. A singleton is a biterminal where one of the tokens is the empty string (ϵ). Neither STGs nor SDTGs are intuitively useful in translating natural languages, since STGs have no way to model reordering, and SDTGs require exponential time to be induced from examples (parallel corpora). Since

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 173–177,
Uppsala, Sweden, 11-16 July 2010.

compilers in general work on well defined, manually specified programming languages, there is no need to induce them from examples, so the exponential complexity is not a problem in this setting – SDTGs can transduce in $\mathcal{O}(n^3)$ time, so once the grammar is known they can be used to translate efficiently.

In natural language translation, the grammar is generally not known, in fact, state-of-the art translation systems rely heavily on machine learning. For transduction grammars, this means that they have to be induced from parallel corpora.

An inversion transduction grammar (ITG) strikes a good balance between STGs and SDTGs, as it allows some reordering, while requiring only polynomial time to be induced from parallel corpora. The allowed reordering is either the identity permutation of the production, or the inversion permutation. Restricting the permutations in this way ensures that an ITG can be expressed in two-normal form, which is the key property for avoiding exponential time complexity in biparsing (parsing of a sentence pair).

An ITG in two-normal form (representing the transduction between L_1 and L_2) is written with identity productions in square brackets, and inverted productions in angle brackets. Each such rule can be construed to represent two (one L_1 and one L_2) synchronized CFG rules:

$$\begin{array}{ccc}
\text{ITG}_{L_1,L_2} & \text{CFG}_{L_1} & \text{CFG}_{L_2} \\
A \rightarrow [\, B\ C \,] & A \rightarrow B\ C & A \rightarrow B\ C \\
A \rightarrow \langle\, B\ C \,\rangle & A \rightarrow B\ C & A \rightarrow C\ B \\
A \rightarrow e/f & A \rightarrow e & A \rightarrow f
\end{array}$$

Inducing an ITG from a parallel corpus is still slow, as the time complexity is $\mathcal{O}(Gn^6)$. Several ways to get around this has been proposed (Zhang et al., 2008; Haghighi et al., 2009; Saers et al., 2009; Saers et al., 2010).

Taking a closer look at the linear ITGs (Saers et al., 2010), there are five rules in normal form. Decomposing these five rule types into monolingual rule types reveals that the monolingual grammars are linear grammars (LGs):

$$\begin{array}{ccc}
\text{LITG}_{L_1,L_2} & \text{LG}_{L_1} & \text{LG}_{L_2} \\
A \rightarrow [\, e/f\ C \,] & A \rightarrow e\ C & A \rightarrow f\ C \\
A \rightarrow [\, B\ e/f \,] & A \rightarrow B\ e & A \rightarrow B\ f \\
A \rightarrow \langle\, e/f\ C \,\rangle & A \rightarrow e\ C & A \rightarrow C\ f \\
A \rightarrow \langle\, B\ e/f \,\rangle & A \rightarrow B\ e & A \rightarrow f\ B \\
A \rightarrow \epsilon/\epsilon & A \rightarrow \epsilon & A \rightarrow \epsilon
\end{array}$$

This means that LITGs are transduction grammars that transduce between linear languages.

There is also a nice parallel in search time complexities between CFGs and ITGs on the one hand, and LGs and LITGs on the other. Searching for all possible parses given a sentence is $\mathcal{O}(n^3)$ for CFGs, and $\mathcal{O}(n^2)$ for LGs. Searching for all possible biparses given a bisentence is $\mathcal{O}(n^6)$ for ITGs, and $\mathcal{O}(n^4)$ for LITGs. This is consistent with thinking of biparsing as finding every L_2 parse for every L_1 parse. Biparsing consists of assigning a joint structure to a sentence pair, rather than assigning a structure to a sentence.

In this paper, only stochastic bracketing grammars (SBITGs and SBLITGs) were used. A bracketing grammar has only one nonterminal symbol, denoted X. A stochastic grammar is one where each rule is associated with a probability, such that

$$\forall X \left[\sum_{\phi} p(X \rightarrow \phi) = 1 \right]$$

While training a Stochastic Bracketing ITG (SBITG) or LITG (SBLITG) with EM, expectations of probabilities over the biparse-forest are calculated. These expectations approach the true probabilities, and can be used as approximations. The probabilities over the biparse-forest can be used to select the one-best parse-tree, which in turn forces an alignment over the sentence pair. The alignments given by SBITGs and SBLITGs has been shown to give better translation quality than bidirectional IBM-models, when applied to short sentence corpora (Saers and Wu, 2009; Saers et al., 2009; Saers et al., 2010). In this paper we explore whether this hold for SBLITGs on standard sentence corpora.

3 Setup

The baseline system for the shared task was a phrase based translation model based on bidirectional IBM- (Brown et al., 1993) and HMM-models (Vogel et al., 1996) combined with the `grow-diag-final-and` heuristic. This is computed with the GIZA++ tool (Och and Ney, 2003) and the Moses toolkit (Koehn et al., 2007). The language model was a 5-gram SRILM (Stolcke, 2002). Parameters in the final translation system were determined with Minimum Error-Rate Training (Och, 2003), and translation quality was assessed with the automatic measures BLEU (Papineni et al., 2002) and NIST (Doddington, 2002).

Corpus	Type	Size
German–English Europarl	out of domain	1,219,343 sentence pairs
German–English news commentary	in-domain	86,941 sentence pairs
English news commentary	in-domain	48,653,884 sentences
German–English news commentary	in-domain tuning data	2,051 sentence pairs
German–English news commentary	in-domain test data	2,489 sentence pairs

Table 1: Corpora available for the German–English translation task after baseline cleaning.

System	BLEU	NIST
GIZA++	**17.88**	5.9748
SBLITG	17.61	5.8846
SBLITG (only Europarl)	17.46	5.8491
SBLITG (only news)	15.49	5.4987
GIZA++ and SBLITG	17.66	5.9650
GIZA++ and SBLITG (only Europarl)	17.58	**5.9819**
GIZA++ and SBLITG (only news)	17.48	5.9693

Table 2: Results for the German–English translation task.

We chose to focus on the German–English translation task. The corpora resources available for that task is summarized in Table 1. We used the entire news commentary monolingual data concatenated with the English side of the Europarl bilingual data to train the language model. In retrospect, this was probably a bad choice, as others seem to prefer the use of two language models instead.

We contrasted the baseline system with pure SBLITG systems trained on different parts of the training data, as well as combined systems, where the SBLITG systems were combined with the baseline system. The combination was done by adding the SBLITG translation model as a second translation path to the base line system.

To train our SBLITG systems, we used the algorithm described in Saers et al. (2010). We set the beam size parameter to 50, and ran expectation-maximization for 10 iterations or until the log-probability of the training corpus started deteriorating. After the grammar was induced we obtained the one-best parse for each sentence pair, which also dictates a word alignment over that sentence pair, which we used instead of the word alignments provided by GIZA++. From that point, training did not differ from the baseline procedure.

We trained a total of three pure SBLITG system, one with only the news commentary part of the corpus, one with only the Europarl part, and one with both. We also combined all three SBLITG systems with the baseline system to see whether the additional translation paths would help.

The system we submitted corresponds to the "GIZA++ and SBLITG (only news)" system, but with RandLM (Talbot and Osborne, 2007) as language model rather than SRILM. This was because we lacked the necessary RAM resources to calculate the full SRILM model before the system submission deadline.

4 Results

The results for the development test set are summarized in Table 2. The submitted system achieved a BLEU score of 0.1759 and a NIST score of 5.9579 for cased output on this year's test set (these numbers are not comparable to those in Table 2). To our surprise, adding the additional phrases as a second translation path does not seem to help. Instead a small deterioration in BLEU is noted (0.22–0.40 points), whereas the differences in NIST are mixed (-0.0098–+0.0071 points). Over all the variations were very small. The pure SBLITG systems perform consistently below baseline, which could indicate that the grammar class is unable to capture the reorderings found in longer sentence pairs adequately in one parse. The variation between the pure SBLITG systems can be explained by the size of the training data: more data – better quality.

5 Conclusions

We tried to use SBLITGs as word aligners on full size sentences, which has not been done to date, and noted that the formalism seems unable to account for the full complexity of longer sentence pairs. We also tried combining the translation models acquired with SBLITG alignments to the baseline system, and noted very small differences, tending to a deterioration in quality. The fact that SBLITGs seem unable to capture the complex relationship between an English and a German sentence in one parse means that we need to find either some more complex model or some way to use the entire parse forest to arrive at the alignment.

Acknowledgments

This work was funded by the Swedish National Graduate School of Language Technology (GSLT), the Defense Advanced Research Projects Agency (DARPA) under GALE Contract No. HR0011-06-C-0023, and the Hong Kong Research Grants Council (RGC) under research grants GRF621008, DAG03/04.EG09, RGC6256/00E, and RGC6083/99E. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency. The computations were performed on UPPMAX resources under project p2007020.

References

Jeffrey D. Aho, Alfred V. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. Prentice-Halll, Englewood Cliffs, NJ.

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.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology conference (HLT-2002)*, San Diego, California.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 923–931, Suntec, Singapore, August.

Philipp Koehn, 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. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

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

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 *41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, Jul.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July.

Markus Saers and Dekai Wu. 2009. Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*, pages 28–36, Boulder, Colorado, June.

Markus Saers, Joakim Nivre, and Dekai Wu. 2009. Learning Stochastic Bracketing Inversion Transduction Grammars with a cubic time biparsing algorithm. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 29–32, Paris, France, October.

Markus Saers, Joakim Nivre, and Dekai Wu. 2010. Word alignment with Stochastic Bracketing Linear Inversion Transduction Grammar. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, June.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *International Conference on Spoken Language Processing*, Denver, Colorado, September.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 512–519, Prague, Czech Republic, June.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841, Morristown, New Jersey.

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

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of noncompositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June.