Word Alignment with Stochastic Bracketing Linear Inversion Transduction Grammar

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

The class of Linear Inversion Transduction Grammars (LITGs) is introduced, and used to induce a word alignment over a parallel corpus. We show that alignment via Stochastic Bracketing LITGs is considerably faster than Stochastic Bracketing ITGs, while still yielding alignments superior to the widely-used heuristic of intersecting bidirectional IBM alignments. Performance is measured as the translation quality of a phrase-based machine translation system built upon the word alignments, and an improvement of 2.85 BLEU points over baseline is noted for French–English.

1 Introduction

Machine translation relies heavily on word alignments, which are usually produced by training IBM-models (Brown et al., 1993) in both directions and combining the resulting alignments via some heuristic. Automatically training an Inversion Transduction Grammar (ITG) has been suggested as a viable way of producing superior alignments (Saers and Wu, 2009). The main problem of using Bracketing ITGs for alignment is that exhaustive biparsing runs in $\mathcal{O}(n^6)$ time. Several ways to lower the complexity of ITGs has been suggested, but in this paper, a different approach is taken. Instead of using full ITGs, we explore the possibility of subjecting the grammar to a linear constraint, making exhaustive biparsing of a sentence pair in $\mathcal{O}(n^4)$ time possible. This can be further improved by applying pruning.

2 Background

A transduction is the bilingual version of a language. A language (L_l) can be formally viewed as a set of sentences, sequences of tokens taken from a specified vocabulary (V_l). A transduction ($T_{e,f}$) between two languages (L_e and L_f) is then a set of sentence pairs, sequences of bitokens from the cross production of the vocabularies of the two languages being transduced ($V_{e,f} = V_e \times V_f$). This adds an extra layer of complexity to finding transductions from raw bitexts, as an alignment has to be imposed.

Simple (STG) and Syntax Directed (SDTG) Transduction Grammars (Aho and Ullman, 1972) can be used to parse transductions between context-free languages. Both work fine as long as a grammar is given and parsing is done as transduction, that is: a sentence in one language is rewritten into the other language. In NLP, interest has shifted away from hand-crafted grammars, towards stochastic grammars induced from corpora. To induce a stochastic grammar from a parallel corpus, expectations of all possible parses over a sentence pair are typically needed. STGs can biparse sentence pairs in polynomial time, but are unable to account for the complexities typically found in natural languages. SDTGs do account for the complexities in natural languages, but are intractable for biparsing.

Inversion transductions (Wu, 1995; Wu, 1997) are a special case of transductions that are not monotone, but where permutations are severely limited. By limiting the possible permutations, biparsing becomes tractable. This in turn means that ITGs can be induced from parallel corpora in polynomial time,

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 341–344,
Los Angeles, California, June 2010.

as well as account for most of the reorderings found between natural languages.

An Inversion transduction is limited so that it must be expressible as non-overlapping groups, internally permuted either by the identity permutation or the inversion permutation (hence the name). This requirement also means that the grammar is binarizable, yielding a two-normal form. A production with the identity permutation is written inside square brackets, while productions with the inversion permutation is written inside angled brackets. This gives us a two-normal form that looks like this (where e/f is a biterminal):

$$A \rightarrow [B\ C]$$
$$A \rightarrow \langle B\ C \rangle$$
$$A \rightarrow e/f$$

The time complexity for exhaustive ITG biparsing is $\mathcal{O}(Gn^6)$, which is typically too large to be applicable to large grammars and long sentence. The grammar constant G can be eliminated by limiting the grammar to a bracketing ITG (BITG), which only has one nonterminal symbol. Saers & Wu (2009) show that it is possible to apply exhaustive biparsing to a large parallel corpus ($\sim 100,000$ sentence pairs) of short sentences (≤ 10 tokens in both language). The word alignments read off the Viterbi parse also increased translation quality when used instead of the alignments from bidirectional IBM alignments.

The $\mathcal{O}(n^6)$ time complexity is somewhat prohibitive for large corpora, so pruning in some form is needed. 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. Haghighi et. al (2009) use a simpler aligner as guidance for pruning, which reduce the time complexity by two orders of magnitude, and also introduce block ITG, which gives many-to-one instead of one-to-one alignments. 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 $\mathcal{O}(n^3)$.

In this paper we introduce Linear ITG (LITG), and apply it to a word-alignment task which is evaluated by the phrase-based statistical machine translation (PBSMT) system that can be built from that.

3 Stochastic Bracketing Linear Inversion Transduction Grammar

A Bracketing Linear Inversion Transduction Grammar (BLITG) is a BITG where rules may have at most one nonterminal symbol in their production. This gives us a normal form that is somewhat different from the usual ITG:

$$X \rightarrow [Xe/f]$$
$$X \rightarrow [e/fX]$$
$$X \rightarrow \langle Xe/f \rangle$$
$$X \rightarrow \langle e/fX \rangle$$
$$X \rightarrow \epsilon/\epsilon$$

where one but not both of the tokens in the biterminal may be the empty string ϵ, if a nonterminal is produced. By associating each rule with a probability, we get a Stochastic BLITG (SBLITG).

3.1 Biparsing Algorithm

The sentence pair to be biparsed consists of two vectors of tokens ($\mathbf{e}$ and $\mathbf{f}$). An item is represented as a nonterminal (X), and one span in each of the languages ($e_{s..t}$ and $f_{u..v}$). For notational convenience, an item will be written as the nonterminal with the spans as subscripts ($X_{s,t,u,v}$). The length of an item is defined as the sum of the length of the two spans: $|X_{s,t,u,v}| = t - s + v - u$. Items are gathered in buckets, B_n, according to their length so that $X_{s,t,u,v} \in B_{|X_{s,t,u,v}|}$. The algorithm is initialized with the item spanning the entire sentence pair:

$$X_{0,|\mathbf{e}|,0,|\mathbf{f}|} \in B_{|X_{0,|\mathbf{e}|,0,|\mathbf{f}|}|}$$

Starting from this top bucket, buckets are processed in larger to smaller order: $B_n, B_{n-1}, \ldots, B_1$. While processing a bucket, only smaller items are added, meaning that B_0 is fully constructed by the time B_1 has been processed. Each item in B_0 can have the rule $X \rightarrow \epsilon/\epsilon$ applied to it, eliminating the nonterminal and halting processing. If there are no items in B_0, parsing has failed.

To process a bucket, each item is extended by all applicable rules, and the nonterminals in the productions are added to their respective buckets.

System	BLEU	NIST	Phrases
GIZA++ (`intersect`)	0.2629	6.7968	146,581,109
GIZA++ (`grow-diag-final`)	0.2632	6.7410	1,298,566
GIZA++ (`grow-diag-final-and`)	0.2742	6.9228	7,340,369
SBLITG ($b = 25$)	**0.3027**	**7.3664**	13,551,915
SBLITG ($b = \infty$)	0.3008	7.3303	12,673,361

Table 1: Results for French–English.

$$
\begin{aligned}
X_{s,t,u,v} \rightarrow\ & \\
& [e_{s,s+1}/f_{u,u+1}\ X_{s+1,t,u+1,v}] \\
\mid\ & [X_{s,t-1,u,v-1}\ e_{t-1,t}/f_{v-1,v}] \\
\mid\ & \langle e_{s,s+1}/f_{v-1,v}\ X_{s+1,t,u,v-1}\rangle \\
\mid\ & \langle X_{s,t-1,u+1,v}\ e_{t-1,t}/f_{u,u+1}\rangle \\
\mid\ & [e_{s,s+1}/\epsilon\ X_{s+1,t,u,v}] \mid \langle e_{s,s+1}/\epsilon\ X_{s+1,t,u,v}\rangle \\
\mid\ & [\epsilon/f_{u,u+1}\ X_{s,t,u+1,v}] \mid \langle X_{s,t,u+1,v}\ \epsilon/f_{u,u+1}\rangle \\
\mid\ & [X_{s,t-1,u,v}\ e_{t-1,t}/\epsilon] \mid \langle X_{s,t-1,u,v}\ e_{t-1,t}/\epsilon\rangle \\
\mid\ & [X_{s,t,u,v-1}\ \epsilon/f_{v-1,v}] \mid \langle \epsilon/f_{v-1,v}\ X_{s,t,u,v-1}\rangle
\end{aligned}
$$

Note that there are two productions on each of the four last rows. These are distinct rules, but the symbols in the productions are identical. This phenomenon is due to the fact that the empty symbols can be "read" off either end of the span. In our experiments, such rules were merged into their non-inverting form, effectively eliminating the last four inverted rules (productions enclosed in angled brackets) above.

3.2 Analysis

Let n be the length of the longer sentence in the pair. The number of buckets will be $\mathcal{O}(n)$, since the longest item will be at most $2n$ long. Within a bucket, there can be $\mathcal{O}(n^2)$ starting points for items, but once the length of one of the spans is fixed, the length of the other follows, adding a factor $\mathcal{O}(n)$, making the total number of items in a bucket $\mathcal{O}(n^3)$. Each item in a bucket can be analyzed in 8 possible ways, requiring $\mathcal{O}(1)$ time. In summary, we have: $\mathcal{O}(n) \times \mathcal{O}(n^3) \times \mathcal{O}(1) = \mathcal{O}(n^4)$

The pruning scheme works by limiting the number of items that are processed from each bucket, reducing the cost of processing a bucket from $\mathcal{O}(n^3)$ to $\mathcal{O}(b)$, where b is the beam width. This gives time complexity $\mathcal{O}(n) \times \mathcal{O}(b) \times \mathcal{O}(1) = \mathcal{O}(bn)$.

4 Experiments

We used the guidelines of the shared task of WMT'08[1] to train our baseline system as well as our experimental system. This includes induction of word alignments with GIZA++ (Och and Ney, 2003), induction of a Phrase-based SMT system (Koehn et al., 2007), and tuning with minimum error rate training (Och, 2003), as well as applying some utility scripts provided for the workshop. The translation model is combined with a 5-gram language model (Stolcke, 2002).

Our experimental system uses alignments from the Viterbi parses, extracted during EM training of an SBLITG on the training corpus, instead of GIZA++. Since EM will converge fairly slowly, it was limited to 10 iterations, after which it was halted.

We used the French–English part of the WMT'08 shared task, but limited the training set to sentence pairs where both sentences were of length 20 or less. This was necessary in order to carry out exhaustive search in the SBLITG algorithm. In total, we had 381,780 sentence pairs for training, and 2,000 sentence pairs each for tuning and testing. The language model was trained with the entire training set.

To evaluate the systems we used BLEU (Papineni et al., 2002) and NIST (Doddington, 2002)

Results are presented in Table 1. It is interesting to note that there is no correlation between the number of phrases extracted and translation quality. The only explanation for the results we are seeing is that the SBLITGs find *better* phrases. Since the only difference is the word alignment strategy, this suggests that the word alignments from SBLITGs are better suited for phrase extraction than those from bidirectional IBM-models. The fact that SBLITGs extract more phrases than bidirectional IBM-models under

[1] `http://www.statmt.org/wmt08/`

the `grow-diag-x` heuristics is significant, since more phrases means that more translation possibilities are extracted. The fact that SBLITGs extract fewer phrases than bidirectional IBM-models under the `intersect` heuristic is also significant, since it implies that simply adding more phrases is a bad strategy. Combined, the two observations leads us to believe that there are some alignments missed by the bidirectional IBM-models that are found by the SBLITG-models. It is also interesting to see that the pruned version outperforms the exhaustive version. We believe this to be because the pruned version approaches the correct grammar faster than the exhaustive. That would mean that the exhaustive SBLITG would be better in the limit, but the experiment was limited to 10 iterations.

5 Conclusion

In this paper we have focused on the benefits of applying SBLITGs to the task of inducing word alignments, which leads to a 2.85 BLEU points improvement compared to the standard model (heuristically combined bidirectional IBM-models). In the future, we hope that LITGs will be a spring board towards full ITGs, with more interesting nonterminals than the BITGs seen in the literature so far. With the possibility of inducing full ITG from parallel corpora it becomes viable to use ITG decoders directly as machine translation systems.

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 DARPA.The computations were performed on UPPMAX resources under project p2007020.

References

A. V. Aho and J. D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. Prentice-Halll, Englewood Cliffs, New Jersey.

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.

G. 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.

A. Haghighi, J. Blitzer, J. DeNero, and D. Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of ACL/IJCNLP 2009*, pages 923–931, Singapore.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Session*, pages 177–180, Prague, Czech Republic.

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

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, Sapporo, Japan.

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

M. Saers and D. Wu. 2009. Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars. In *Proceedings of SSST-3 at NAACL HLT 2009*, pages 28–36, Boulder, Colorado.

M. Saers, J. Nivre, and D. Wu. 2009. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *Proceedings of IWPT'09*, pages 29–32, Paris, France.

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

D. Wu. 1995. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proceedings of WVLC-3*, pages 69–82, Cambridge, Massachusetts.

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

H. Zhang, C. Quirk, R. C. Moore, and D. Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL/HLT 2008*, pages 97–105, Columbus, Ohio.