Dealing with Spurious Ambiguity in Learning ITG-based Word Alignment

Shujian Huang
State Key Laboratory for Novel Software Technology
Nanjing University
huangsj@nlp.nju.edu.cn

Stephan Vogel
Language Technologies Institute
Carnegie Mellon University
vogel@cs.cmu.edu

Jiajun Chen
State Key Laboratory for Novel Software Technology
Nanjing University
chenjj@nlp.nju.edu.cn

Abstract

Word alignment has an exponentially large search space, which often makes exact inference infeasible. Recent studies have shown that inversion transduction grammars are reasonable constraints for word alignment, and that the constrained space could be efficiently searched using synchronous parsing algorithms. However, spurious ambiguity may occur in synchronous parsing and cause problems in both search efficiency and accuracy. In this paper, we conduct a detailed study of the causes of spurious ambiguity and how it effects parsing and discriminative learning. We also propose a variant of the grammar which eliminates those ambiguities. Our grammar shows advantages over previous grammars in both synthetic and real-world experiments.

1 Introduction

In statistical machine translation, word alignment attempts to find word correspondences in parallel sentence pairs. The search space of word alignment will grow exponentially with the length of source and target sentences, which makes the inference for complex models infeasible (Brown et al., 1993). Recently, inversion transduction grammars (Wu, 1997), namely ITG, have been used to constrain the search space for word alignment (Zhang and Gildea, 2005; Cherry and Lin, 2007; Haghighi et al., 2009; Liu et al., 2010). ITG is a family of grammars in which the right hand side of the rule is either two nonterminals or a terminal sequence. The most general case of the ITG family is the bracketing transduction grammar (BTG, Figure 1), which has only one nonterminal symbol.

$$A \rightarrow [AA] \mid \langle AA \rangle \mid e/f \mid \epsilon/f \mid e/\epsilon$$

Figure 1: BTG rules. $[AA]$ denotes a monotone concatenation and $\langle AA \rangle$ denotes an inverted concatenation.

Synchronous parsing of ITG may generate a large number of different derivations for the same underlying word alignment. This is often referred to as the spurious ambiguity problem. Calculating and saving those derivations will slow down the parsing speed significantly. Furthermore, spurious derivations may fill up the n-best list and supersede potentially good results, making it harder to find the best alignment. Besides, over-counting those spurious derivations will also affect the likelihood estimation. In order to reduce spurious derivations, Wu (1997), Haghighi et al. (2009), Liu et al. (2010) propose different variations of the grammar. These grammars have different behaviors in parsing efficiency and accuracy, but so far no detailed comparison between them has been done.

In this paper, we formally analyze alignments under ITG constraints and the different causes of spurious ambiguity for those alignments. We do an empirical study of the influence of spurious ambiguity on parsing and discriminative learning by comparing different grammars in both synthetic and real-data experiments. To our knowledge, this is the first in-depth analysis on this specific issue. A new variant of the grammar is proposed, which efficiently removes all spurious ambiguities. Our grammar shows advantages over previous ones in both experiments.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 379–383,
Portland, Oregon, June 19-24, 2011.

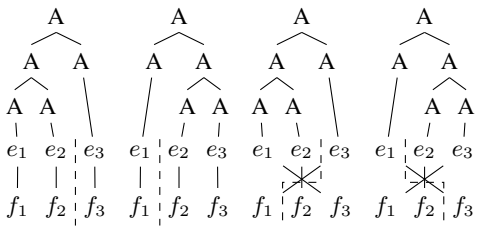

Figure 2: Possible monotone/inverted t-splits (dashed lines) under BTG, causing branching ambiguities.

2 ITG Alignment Family

By lexical rules like $A \rightarrow e/f$, each ITG derivation actually represents a unique alignment between the two sequences. Thus the family of ITG derivations represents a family of word alignment.

Definition 1. The *ITG alignment family* is a set of word alignments that has at least one BTG derivation.

ITG alignment family is only a subset of word alignments because there are cases, known as inside-outside alignments (Wu, 1997), that could not be represented by any ITG derivation. On the other hand, an ITG alignment may have multiple derivations.

Definition 2. For a given grammar G, *spurious ambiguity* in word alignment is the case where two or more derivations d_1, d_2, ... d_k of G have the same underlying word alignment A. A grammar G is *non-spurious* if for any given word alignment, there exist at most one derivation under G.

In any given derivation, an ITG rule applies by either generating a bilingual word pair (lexical rules) or splitting the current alignment into two parts, which will recursively generate two sub-derivations (transition rules).

Definition 3. Applying a monotone (or inverted) concatenation transition rule forms a *monotone t-split* (or *inverted t-split*) of the original alignment (Figure 2).

3 Causes of Spurious Ambiguity

3.1 Branching Ambiguity

As shown in Figure 2, left-branching and right-branching will produce different derivations under BTG, but yield the same word alignment. Branching ambiguity was identified and solved in Wu (1997), using the grammar in Figure 3, denoted as LG. LG uses two separate non-terminals for monotone and inverted concatenation, respectively. It only allows left branching of such non-terminals, by excluding rules like $A \rightarrow [BA]$.

$$A \rightarrow [AB] \mid [BB] \mid [CB] \mid [AC] \mid [BC] \mid [CC]$$
$$B \rightarrow \langle AA \rangle \mid \langle BA \rangle \mid \langle CA \rangle \mid \langle AC \rangle \mid \langle BC \rangle \mid \langle CC \rangle$$
$$C \rightarrow e/f \mid \epsilon/f \mid e/\epsilon$$

Figure 3: A Left heavy Grammar (LG).

Theorem 1. For each ITG alignment A, in which all the words are aligned, LG will produce a unique derivation.

Proof: Induction on n, the length of A. Case n=1 is trivial. Induction hypothesis: the theorem holds for any A with length less than n.

For A of length n, let s be the right most t-split which splits A into S_1 and S_2. s exists because A is an ITG alignment. Assume that there exists another t-split s', splitting A into S_{11} and $(S_{12}S_2)$. Because A is fixed and fully aligned, it is easy to see that if s is a monotone t-split, s' could only be monotone, and S_{12} and S_2 in the right sub-derivation of t-split s' could only be combined by monotone concatenation as well. So s' will have a right branching of monotone concatenation, which contradicts with the definition of LG because right branching of monotone concatenations is prohibited. A similar contradiction occurs if s is an inverted t-split. Thus s should be the unique t-split for A. By I.H., S_1 and S_2 have a unique derivation, because their lengths are less than n. Thus the derivation for A will be unique.

3.2 Null-word Attachment Ambiguity

Definition 4. For any given sentence pair (e, f) and its alignment A, let (e', f') be the sentence pairs with all null-aligned words removed from (e, f). The *alignment skeleton* A_S is the alignment between (e', f') that preserves all links in A.

From Theorem 1 we know that every ITG alignment has a unique LG derivation for its alignment skeleton (Figure 4 (c)).

However, because of the lexical or syntactic differences between languages, some words may have

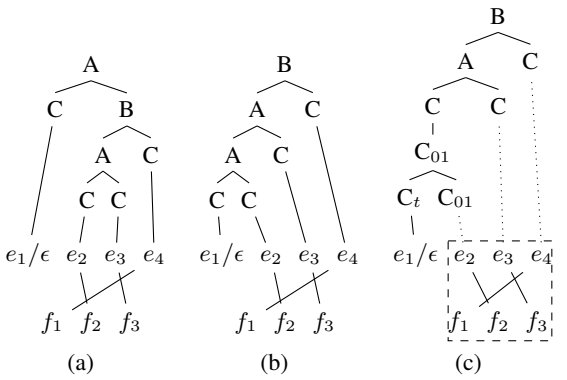

Figure 4: Null-word attachment for the same alignment. ((a) and (b) are spurious derivations under LG caused by null-aligned words attachment. (c) shows the unique derivation under LGFN. The dotted lines have omitted some unary rules for simplicity. The dashed box marks the alignment skeleton.)

$$A \rightarrow [AB] \mid [BB] \mid [CB] \mid [AC] \mid [BC] \mid [CC]$$
$$B \rightarrow \langle AA \rangle \mid \langle BA \rangle \mid \langle CA \rangle \mid \langle AC \rangle \mid \langle BC \rangle \mid \langle CC \rangle$$
$$C \rightarrow C_{01} \mid [C_s\ C]$$
$$C_{01} \rightarrow C_{00} \mid [C_t\ C_{01}]$$
$$C_{00} \rightarrow e/f,\ C_t \rightarrow e/\epsilon,\ C_s \rightarrow \epsilon/f$$

Figure 5: A Left heavy Grammar with Fixed Null-word attachment (LGFN).

no explicit correspondence in the other language and tend to stay unaligned. These null-aligned words, also called singletons, should be attached to some other nodes in the derivation. It will produce different derivations if those null-aligned words are attached by different rules, or to different nodes.

Haghighi et al. (2009) give some restrictions on null-aligned word attachment. However, they fail to restrict the node to which the null-aligned word is attached, e.g. the cases (a) and (b) in Figure 4.

3.3 LGFN Grammar

We propose here a new variant of ITG, denoted as LGFN (Figure 5). Our grammar takes similar transition rules as LG and efficiently constrains the attachment of null-aligned words. We will empirically compare those different grammars in the next section.

Lemma 1. LGFN has a unique mapping from the derivation of any given ITG alignment A to the derivation of its alignment skeleton A_S.

Proof: LGFN maps the null-aligned source word sequence, $C_{s_1}, C_{s_2}, ..., C_{s_k}$, the null-aligned target word sequence, $C_{t_1}, C_{t_2}, ..., C_{t_{k'}}$, together with the aligned word-pair C_{00} that directly follows, to the node C exactly in the way of Equation 1. The brackets indicate monotone concatenations.

$$C \rightarrow [C_{s_1}...[C_{s_k}[C_{t_1}...[C_{t_{k'}}C_{00}]...]]...] \quad (1)$$

The mapping exists when every null-aligned sequence has an aligned word-pair after it. Thus it requires an artificial word at the end of the sentence.

Note that our grammar attaches null-aligned words in a right-branching manner, which means it builds the span only when there is an aligned word-pair. After initialization, any newly-built span will contain at least one aligned word-pair. Comparatively, the grammar in Liu et al. (2010) uses a left-branching manner. It may generate more spans that only contain null-aligned words, which makes it less efficient than ours.

Theorem 2. LGFN has a unique derivation for each ITG alignment, i.e. LGFN is non-spurious.

Proof: Derived directly from Definition 4, Theorem 1 and Lemma 1.

4 Experiments

4.1 Synthetic Experiments

We automatically generated 1000 fully aligned ITG alignments of length 20 by generating random permutations first and checking ITG constraints using a linear time algorithm (Zhang et al., 2006). Sparser alignments were generated by random removal of alignment links according to a given null-aligned word ratio. Four grammars were used to parse these alignments, namely LG (Wu, 1997), HaG (Haghighi et al., 2009), LiuG (Liu et al., 2010) and LGFN (Section 3.3).

Table 1 shows the average number of derivations per alignment generated under LG and HaG. The number of derivations produced by LG increased dramatically because LG has no restrictions on null-aligned word attachment. HaG also produced a large number of spurious derivations as the number of null-aligned words increased. Both LiuG and LGFN produced a unique derivation for each alignment, as expected. One interpretation is that in order to get

%	0	5	10	15	20	25
LG	1	42.2	1920.8	9914.1+	10000+	10000+
HaG	1	3.5	10.9	34.1	89.2	219.9

Table 1: Average #derivations per alignment for LG and HaG v.s. Percentage of unaligned words. (+ marked parses have reached the beam size limit of 10000.)

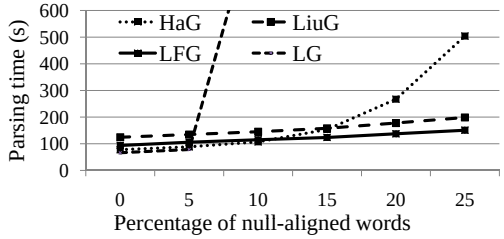

Figure 6: Total parsing time (in seconds) v.s. Percentage of un-aligned words.

the 10-best alignments for sentence pairs that have 10% of words unaligned, the top 109 HaG derivations should be generated, while the top 10 LiuG or LGFN derivations are already enough.

Figure 6 shows the total parsing time using each grammar. LG and HaG showed better performances when most of the words were aligned because their grammars are simpler and less constrained. However, when the number of null-aligned words increased, the parsing times for LG and HaG became much longer, caused by the calculation of the large number of spurious derivations. Parsings using LG for 10 and 15 percent of null-aligned words took around 15 and 80 minutes, respectively, which cannot be plotted in the same scale with other grammars. The parsing times of LGFN and LiuG also slowly increased, but parsing LGFN consistently took less time than LiuG.

It should be noticed that the above results came from parsing according to some given alignment. When searching without knowing the correct alignment, it is possible for every word to stay unaligned, which makes spurious ambiguity a much more serious issue.

4.2 Discriminative Learning Experiments

To further study how spurious ambiguity affects the discriminative learning, we implemented a framework following Haghighi et al. (2009). We used a log-linear model, with features like IBM model1

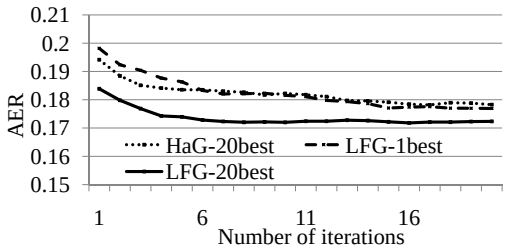

Figure 7: Test set AER after each iteration.

probabilities (collected from FBIS data), relative distances, matchings of high frequency words, matchings of pos-tags, etc. Online training was performed using the margin infused relaxed algorithm (Crammer et al., 2006), MIRA. For each sentence pair (e, f), we optimized with alignment results generated from the nbest parsing results. Alignment error rate (Och and Ney, 2003), AER, was used as the loss function. We ran MIRA training for 20 iterations and evaluated the alignments of the best-scored derivations on the test set using the average weights.

We used the manually aligned Chinese-English corpus in NIST MT02 evaluation. The first 200 sentence pairs were used for training, and the last 150 for testing. There are, on average, 10.3% words stay null-aligned in each sentence, but if restricted to sure links the average ratio increases to 22.6%.

We compared training using LGFN with 1-best, 20-best and HaG with 20-best (Figure 7). Training with HaG only obtained similar results with 1-best trained LGFN, which demonstrated that spurious ambiguity highly affected the nbest list here, resulting in a less accurate training. Actually, the 20-best parsing using HaG only generated 4.53 different alignments on average. 20-best training using LGFN converged quickly after the first few iterations and obtained an AER score (17.23) better than other systems, which is also lower than the refined IBM Model 4 result (19.07).

We also trained a similar discriminative model but extended the lexical rule of LGFN to accept at maximum 3 consecutive words. The model was used to align FBIS data for machine translation experiments. Without initializing by phrases extracted from existing alignments (Cherry and Lin, 2007) or using complicated block features (Haghighi et al.,

2009), we further reduced AER on the test set to 12.25. An average improvement of 0.52 BLEU (Papineni et al., 2002) score and 2.05 TER (Snover et al., 2006) score over 5 test sets for a typical phrase-based translation system, Moses (Koehn et al., 2003), validated the effectiveness of our experiments.

5 Conclusion

Great efforts have been made in reducing spurious ambiguities in parsing combinatory categorial grammar (Karttunen, 1986; Eisner, 1996). However, to our knowledge, we give the first detailed analysis on spurious ambiguity of word alignment. Empirical comparisons between different grammars also validates our analysis.

This paper makes its own contribution in demonstrating that spurious ambiguity has a negative impact on discriminative learning. We will continue working on this line of research and improve our discriminative learning model in the future, for example, by adding more phrase level features.

It is worth noting that the definition of spurious ambiguity actually varies for different tasks. In some cases, e.g. bilingual chunking, keeping different null-aligned word attachments could be useful. It will also be interesting to explore spurious ambiguity and its effects in those different tasks.

Acknowledgments

The authors would like to thank Alon Lavie, Qin Gao and the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61003112), the National Fundamental Research Program of China (2010CB327903) and by NSF under the CluE program, award IIS 084450.

References

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, SSST '07, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December.

Jason Eisner. 1996. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aria Haghighi, John Blitzer, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Association for Computational Linguistics*, Singapore.

Lauri Karttunen. 1986. Radical lexicalism. Technical Report CSLI-86-68, Stanford University.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*.

Shujie Liu, Chi-Ho Li, and Ming Zhou. 2010. Discriminative pruning for discriminative itg alignment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 316–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie J. Dorr, and Richard Schwartz. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23:377–403, September.

Hao Zhang and Daniel Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 475–482, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 256–263, Morristown, NJ, USA. Association for Computational Linguistics.