Generalizing Hierarchical Phrase-based Translation using Rules with Adjacent Nonterminals

Hendra Setiawan and Philip Resnik
UMIACS Laboratory for Computational Linguistics and Information Processing
University of Maryland, College Park, MD 20742, USA
hendra, resnik @umd.edu

Abstract

Hierarchical phrase-based translation (Hiero, (Chiang, 2005)) provides an attractive framework within which both short- and long-distance reorderings can be addressed consistently and efficiently. However, Hiero is generally implemented with a constraint preventing the creation of rules with adjacent nonterminals, because such rules introduce computational and modeling challenges. We introduce methods to address these challenges, and demonstrate that rules with adjacent nonterminals can improve Hiero's generalization power and lead to significant performance gains in Chinese-English translation.

1 Introduction

Hierarchical phrase-based translation (Hiero, (Chiang, 2005)) has proven to be a very useful compromise between syntactically informed and purely corpus-driven translation. By automatically learning synchronous grammar rules from parallel text, Hiero captures short- and long-distance reorderings consistently and efficiently. However, implementations of Hiero generally forbid adjacent nonterminal symbols on the source side of hierarchical rules, a practice we will refer to as the *non-adjacent nonterminals constraint*. The main argument against such rules is that they cause the system to produce multiple derivations that all lead to the same translation – a form of redundancy known as *spurious ambiguity*. Spurious ambiguity can lead to drastic reductions in decoding efficiency, and the obvious solutions, such as reducing beam width, erode translation quality.

In Section 2, we argue that the non-adjacent nonterminals constraints severely limits Hiero's generalization power, limiting its coverage of important reordering phenomena. In Section 3, we discuss the challenges that arise in relaxing this constraint. In Section 4 we introduce new methods to address those challenges, and Section 5 validates the approach empirically.

Improving Hiero via variations on rule pruning and filtering is well explored, e.g., (Chiang, 2005; Chiang et al., 2008; Zollmann and Venugopal, 2006), to name just a few. These proposals differ from each other mainly in the specific linguistic knowledge being used, and on which side the constraints are applied. In contrast, we complement previous work by showing that *adding* rules to Hiero can provide benefits if done judiciously.

2 Judicious Use of Adjacent Nonterminals

Our motivations largely follow Menezes and Quirk's (2007) discussion of reorderings and generalization. As a specific example, we will use a Chinese to English verb phrase (VP) translation (Fig. 1), which represents one of the most prominent phrase constructions in Chinese. Here the construction of the Chinese VP involves joining a prepositional phrase (PP) and a smaller verbal phrase (VP-A), with the preposition at the beginning as a PP marker. In the translation, the VP-A precedes the PP, a shift from pre-verbal PP in Chinese to post-verbal in English.

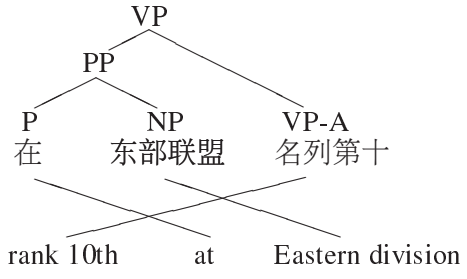

Figure 1: A Chinese-English verb phrase translation

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 349–352,
Los Angeles, California, June 2010.

Hiero can correctly translate the example if it learns any of the following rules from training data:

$X \rightarrow \langle$ 在 X_1 名列第十, rank 10th at $X_1\rangle$ (1)

$X \rightarrow \langle$ 在东部联盟 X_1, X_1 at Eastern div.$\rangle$ (2)

$X \rightarrow \langle X_1$ 东部联盟 X_2, X_2 X_1 Eastern div.$\rangle$ (3)

However, in practice, data sparsity makes the chance of learning these rules rather slim. For instance, learning Rule 1 depends on training data containing instances of the shift with identical wording for the VP-A, which belongs to an open word class.

If Hiero fails to learn any of the above rules, it will apply the "glue rules" $S \rightarrow \langle S\ X_1,\ S\ X_1\rangle$ and $S \rightarrow \langle X,\ X\rangle$. But these glue rules clearly cannot model the VP-A's movement. In failing to learn Rules 1-3, Hiero has no choice but to translate VP-A in a monotone order.

On the other hand, consider the following rules with adjacent nonterminals on the source side (or *XX rules*, for brevity):

$X \rightarrow \langle$在 X_1X_2, X_2 at $X_1\rangle$ (4)

$X \rightarrow \langle X_1X_2$ 名列第十, rank 10th $X_1X_2\rangle$ (5)

$X \rightarrow \langle X_1X_2, X_2X_1\rangle$ (6)

Note that although XX rules 4-6 can potentially increase the chance of modeling the pre-verbal to post-verbal shift, not all of them are beneficial to learn. For instance, Rule 5 models the word order shift but introduces spurious ambiguity, since the nonterminals are translated in monotone order. Rule 6, which resembles the inverted rule of the Inversion Transduction Grammar (Wu, 1997), is highly ambiguous because its application has no lexical grounding. Rule 4 avoids both problems, and is also easier to learn, since it is lexically anchored by a preposition, 在(at), which we can expect to appear frequently in training. These observations will motivate us to focus on rules that model non-monotone reordering of phrases surrounding a lexical item on the target side.

3 Addressing XX Rule Challenges

The first challenge created by introducing XX rules is computational: relaxing the constraint significantly increases the grammar size. Motivated by our earlier discussion, we address this by permitting only rules that model non-monotone reordering, i.e. those rules whose nonterminals are projected into the target language in a different word order, leaving monotone mappings to be handled by the glue rules as previously. This choice helps keep the search space more manageable, and also avoids spurious ambiguity. In addition, we disallow rules in which nonterminals are adjacent on both the source and target sides, by imposing the non adjacent nonterminal constraint on the target side whenever the constraint is relaxed on the source side. This forces any non-monotone reorderings to always be grounded in lexical evidence. We refer to the permitted subset of XX rules as *XX-nonmono* rules.

The second challenge involves modeling: introducing XX rules places them in competition with the existing glue rules. In particular, these two kinds of rules try to model the same phenomena, namely the translations of phrases that appear next to each other. However, they differ in terms of the features associated with the rules. XX rules will be associated with the same features as any other hierarchical rules, since they are all learned via an identical training method. In contrast, glue rules are introduced into the grammar in an *ad hoc* manner, and the only feature associated with them is a "glue penalty". These distinct feature sets makes direct comparison of scores unreliable. As a result the decoder may simply prefer to always select glue rules because they are associated with fewer features resulting in adjacent phrases always being translated in a monotone order. To address this issue, we introduce a new model, which we call the *target-side function words orientation-based model*, or simply P_{ori_t}, which evaluates the application of the two kinds of rules on the same context, i.e. for our example, it is the function word 在(at).

4 Target-side Function Words Orientation-based Model

The P_{ori_t} model is motivated by the *function words reordering hypothesis* (Setiawan et al., 2007), which suggests that function words encode essential information about the (re)ordering of their neighboring phrases. In contrast to Setiawan et al. (2007), who looked at neighboring contexts for function words on the source side, we focus here on modeling the influence of function words on neighboring phrases

on the *target* side. We argue that this focus better fits our purpose, since the phrases that we want to model are the function words' neighbors on the target side, as illustrated in Fig. 1.

To develop this idea, we first define an ori_t function that takes a source function word as a reference point, along with its neighboring phrase on the target side. The ori_t function outputs one of the following orientation values (Nagata et al., 2006): Monotone-Adjacent (MA); Reverse-Adjacent (RA); Monotone-Gap (MG); and Reverse-Gap (RG). The Monotone/Reverse distinction indicates whether the source order follows the target order. The Adjacent/Gap distinction indicates whether the two phrases are adjacent or separated by an intervening phrase on the source side. For example, in Fig. 1, the value of ori_t for right neighbor *Eastern division* with respect to function word 在 (at) is MA, since its corresponding source phrase 东部联盟 is adjacent to 在 (at) and their order is preserved on the English side. The value for left neighbor *rank 10th* with respect to 在 (at) is RG, since 名列第十 is separated from 在 (at) and their order is reversed on the English side.

More formally, we define $P_{ori_t}(ori_t(Y,X)|Y)$, where $ori_t(Y,X) \in \{\text{MA}, \text{RA}, \text{MG}, \text{RG}\}$ is the orientation of a target phrase X with a source function word Y as the reference point.[1]

We estimate the orientation model using maximum likelihood, which involves counting and normalizing events of interest: $(Y, o = ori_t(Y,X))$. Specifically, we estimate $P_{ori_t}(o|Y) = C(Y,o)/C(Y,\cdot)$. Collecting training counts $C(Y,o)$ involves several steps. First, we run GIZA++ on the training bitext and apply the "grow-diag-final" heuristic over the training data to produce a bi-directional word alignment. Then, we enumerate all occurrences of Y and determine $ori_t(Y,X)$. To ensure uniqueness, we enforce that neighbor X be the longest possible phrase that satisfies the consistency constraint (Och and Ney, 2004). Determining $ori_t(Y,X)$ can then be done in a straightforward manner by looking at the monotonicity (monotone or reverse) and adjacency (adjacent or gap) between Y's and X.

[1]In fact, separate models are developed for left and right neighbors, although for clarity we suppress this distinction throughout.

	MT06	MT08
baseline	30.58	23.59
+itg	29.82	23.21
+XX	30.10	22.86
+XX-nonmono	*30.96*	*24.07*
$+ori_t$	30.19	*23.69*
+XX-nonmono+ori_t	***31.49***	***24.73***

Table 1: Experimental results where better than baseline results are *italicized*, and statistically significant better ($p < 0.01$) are in **bold**.

5 Experiments

We evaluated the generalization of Hiero to include XX rules on a Chinese-to-English translation task. We treat the $N = 128$ most frequent words in the corpus as function words, an approximation that has worked well in the past and minimized dependence on language-specific resources (Setiawan et al., 2007). We report BLEU r4n4 and assess significance using the standard bootstrapping approach.

We trained on the NIST MT06 Eval corpus excluding the UN data (approximately 900K sentence pairs), segmenting Chinese using the Harbin segmenter (Zhao et al., 2001). Our 5-gram language model with modified Kneser-Ney smoothing was trained on the English side of our training data plus portions of the Gigaword v2 English corpus. We optimized the feature weights using minimum error rate training, using the NIST MT03 test set as the development set. We report the results on the NIST 2006 evaluation test (MT06) and the NIST 2008 evaluation test (MT08).

Table 1 reports experiments in an incremental fashion, starting from the baseline model (the original Hiero), then adding different sets of rules, and finally adding the orientation-based model. In our first experiments, we investigated the introduction of three different sets of XX rules. First (+itg), we simply add the ITG's inverted rule (Rule 6) to the baseline system in an ad-hoc manner, similar to the glue rules. This hurts performance consistently across MT06 and MT08 sets, which we suspect is a result of ITG rule applications often aggravating search error. Second (+XX), we permitted general XX rules. This results in a grammar size increase of 25-26%, filtering out rules irrelevant for the test set,

and leads to a significant performance drop, again perhaps attributable to search error. When we inspected the rules, we observe that the majority of these rules involve spurious word insertions. Third (+XX-nonmono), we introduced only XX-nonmono rules; this produced only a 5% additional rules, and yielded a marginal but consistent gain.

In a second experiment ($+ori_t$), we introduced the target-side function words orientation-based model. Note that this experiment is orthogonal to the first set, since we introduce no additional rules. Results are mixed, worse for MT06 but better (with significance) for MT08. Here, we suspect the model's potential has not been fully realized, since Hiero only considers monotone reordering in unseen cases.

Finally, we combine both the XX-nonmono rules and the P_{ori_t} model (+XX-nonmono+ori_t). The combination produces a significant, consistent gain across all test sets. This result suggests that the orientation model contributes more strongly in unseen cases when Hiero also considers non-monotone reordering. We interpret this result as a validation of our hypothesis that carefully relaxing the non-adjacent constraint improves translation.

6 Discussion and Future Work

To our knowledge, the work reported here is the first to relax the non-adjacent nonterminals constraint in hierarchical phrase-based models. The results confirm that judiciously *adding* rules to a Hiero grammar, adjusting the modeling accordingly, can achieve significant gains.

Although we found that XX-nonmono rules performed better than general XX rules, we believe the latter may nonetheless prove useful. Manually inspecting our system's output, we find that the output is often shorter than the references, and the missing words often correspond to function words that are modeled by those rules. Using XX rules to model legitimate word insertions is a topic for future work.

Acknowledgments

The authors gratefully acknowledge partial support from the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-2-001. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii, October.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Arul Menezes and Chris Quirk. 2007. Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. 2006. A clustered global phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 713–720, Sydney, Australia, July. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Hendra Setiawan, Min-Yen Kan, and Haizhou Li. 2007. Ordering phrases with function words. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 712–719, Prague, Czech Republic, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep.

Tiejun Zhao, Yajuan Lv, Jianmin Yao, Hao Yu, Muyun Yang, and Fang Liu. 2001. Increasing accuracy of chinese segmentation with strategy of multi-step processing. *Journal of Chinese Information Processing (Chinese Version)*, 1:13–18.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.