Relabeling Syntax Trees to Improve Syntax-Based Machine Translation Quality

Bryant Huang
Language Weaver, Inc.
4640 Admiralty Way, Suite 1210
Marina del Rey, CA 90292
bhuang@languageweaver.com

Kevin Knight
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
knight@isi.edu

Abstract

We identify problems with the Penn Treebank that render it imperfect for syntax-based machine translation and propose methods of relabeling the syntax trees to improve translation quality. We develop a system incorporating a handful of relabeling strategies that yields a statistically significant improvement of 2.3 BLEU points over a baseline syntax-based system.

1 Introduction

Recent work in statistical machine translation (MT) has sought to overcome the limitations of phrase-based models (Marcu and Wong, 2002; Koehn et al., 2003; Och and Ney, 2004) by making use of syntactic information. Syntax-based MT offers the potential advantages of enforcing syntax-motivated constraints in translation and capturing long-distance/non-contiguous dependencies. Some approaches have used syntax at the core (Wu, 1997; Alshawi et al., 2000; Yamada and Knight, 2001; Gildea, 2003; Eisner, 2003; Hearne and Way, 2003; Melamed, 2004) while others have integrated syntax into existing phrase-based frameworks (Xia and McCord, 2004; Chiang, 2005; Collins et al., 2005; Quirk et al., 2005).

In this work, we employ a syntax-based model that applies a series of tree/string (xRS) rules (Galley et al., 2004; Graehl and Knight, 2004) to a source language string to produce a target language phrase structure tree. Figure 1 exemplifies the translation process, which is called a *derivation*, from Chinese into English. The source string to translate (枪手 被 警方 击毙 .) is shown at the top left. Rule ① replaces the Chinese word 警方 (shaded) with the English **NP-C** *police*. Rule ② then builds a **VP** over the 被 **NP-C** 击毙 sequence. Next, 枪手 is translated as the **NP-C** *the gunman* by rule ③. Finally, rule ④ combines the sequence of **NP-C VP** . into an **S**, denoting a complete tree. The yield of this tree gives the target translation: *the gunman was killed by police .*

The Penn English Treebank (PTB) (Marcus et al., 1993) is our source of syntactic information, largely due to the availability of reliable parsers. It is not clear, however, whether this resource is suitable, as is, for the task of MT. In this paper, we argue that the overly-general tagset of the PTB is problematic for MT because it fails to capture important grammatical distinctions that are critical in translation. As a solution, we propose methods of relabeling the syntax trees that effectively improve translation quality.

Consider the derivation in Figure 2. The output translation has two salient errors: determiner/noun number disagreement (**this Turkish positions*) and auxiliary/verb tense disagreement (**has demonstrate*). The first problem arises because the **DT** tag, which does not distinguish between singular and plural determiners, allows singular *this* to be used with plural **NNS** *positions*. In the second problem, the **VP-C** tag fails to communicate that it is headed by the base verb (**VB**) *demonstrate*, which should prevent it from being used with the auxiliary **VBZ** *has*. Information-poor tags like **DT** and **VP-C** can be relabeled to encourage more fluent translations, which is the thrust of this paper.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 240–247,
New York, June 2006.

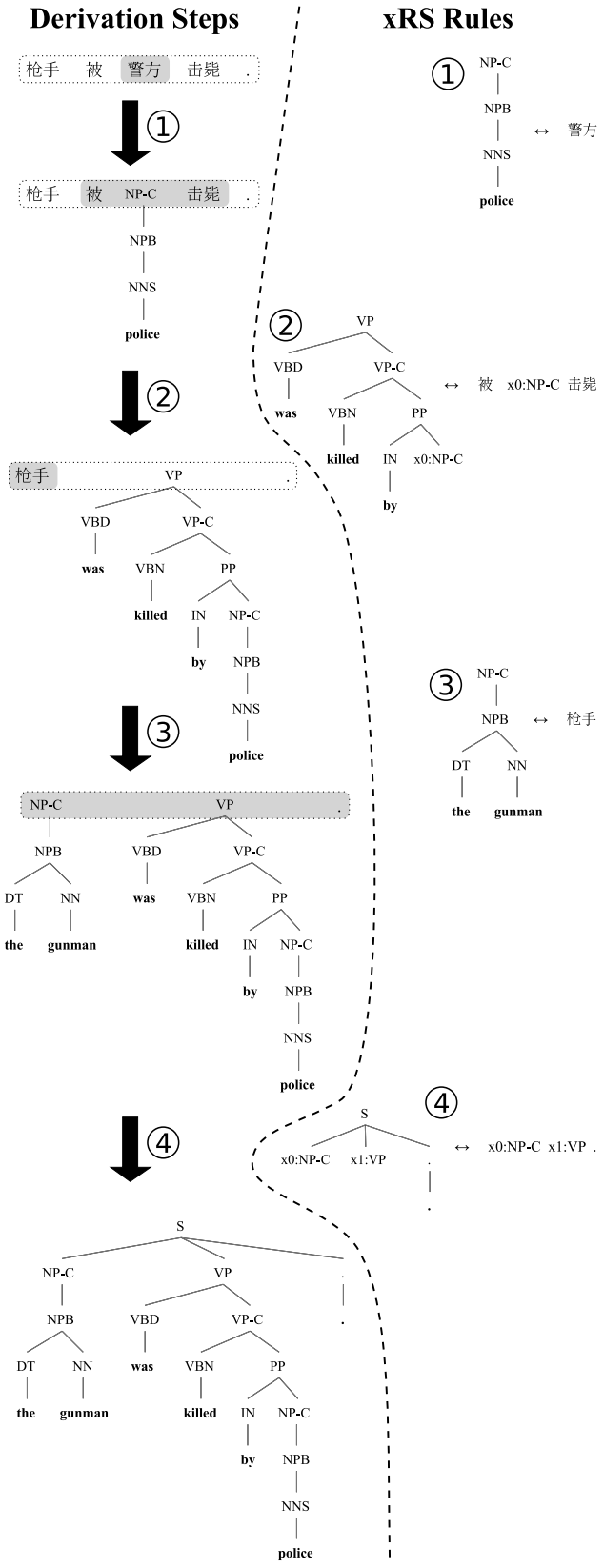

Figure 1: A derivation from a Chinese sentence to an English tree.

Section 2 describes our data and experimental procedure. Section 3 explores different relabeling approaches and their impact on translation quality. Section 4 reports a substantial improvement in BLEU achieved by combining the most effective relabeling methods. Section 5 concludes.

2 Experimental Framework

Our training data consists of 164M+167M words of parallel Chinese/English text. The English half was parsed with a reimplementation of Collins' Model 2 (Collins, 1999) and the two halves were word-aligned using GIZA++ (Och and Ney, 2000). These three components — Chinese strings, English parse trees, and their word alignments — were inputs to our experimental procedure, which involved five steps: (1) tree relabeling, (2) rule extraction, (3) decoding, (4) n-best reranking, (5) evaluation.

This paper focuses on step 1, in which the original English parse trees are transformed by one or more relabeling strategies. Step 2 involves extracting *minimal* xRS rules (Galley et al., 2004) from the set of string/tree/alignments triplets. These rules are then used in a CKY-type parser-decoder to translate the 878-sentence 2002 NIST MT evaluation test set (step 3). In step 4, the output 2,500-sentence n-best list is reranked using an n-gram language model trained on 800M words of English news text. In the final step, we score our translations with 4-gram BLEU (Papineni et al., 2002).

Separately for each relabeling method, we ran these five steps and compared the resulting BLEU score with that of a baseline system with no relabeling. To determine if a BLEU score increase or decrease is meaningful, we calculate statistical significance at 95% using paired bootstrap resampling (Koehn, 2004; Zhang et al., 2004) on 1,000 samples.

Figure 3 shows the results from each relabeling experiment. The second column indicates the change in the number of unique rules from the baseline number of 16.7M rules. The third column gives the BLEU score along with an indication whether it is a statistically significant increase (▲), a statistically significant decrease (▼), or neither (**?**) over the baseline BLEU score.

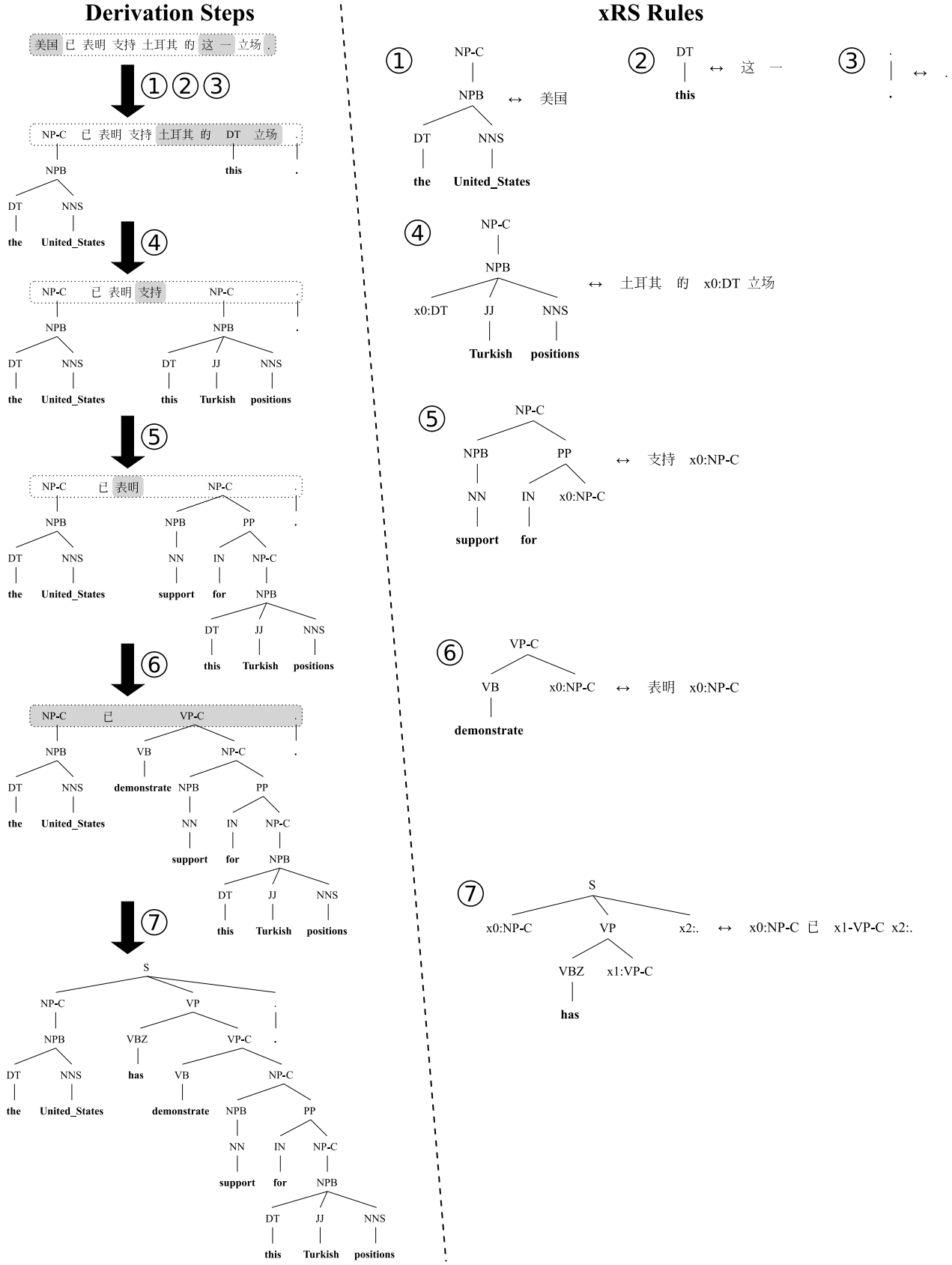

Figure 2: A bad translation fixable by relabeling.

Relabeling	Variant	Δ # Rules	BLEU	Δ
BASELINE		—	20.06	—
LEX_PREP	1	+301.2K	20.2	▲
	2	+254.8K	20.36	▲
	3	+188.3K	20.14	▲
LEX_DT	1	+36.1K	20.15	▲
	2	+29.6K	20.18	▲
LEX_AUX	1	+5.1K	20.09	▲
	2	+8.0K	20.09	?
	3	+1.6K	20.11	▲
	4	+13.8K	20.07	?
LEX_CC		+3.3K	20.03	▼
LEX_%		+0.3K	20.14	▲
TAG_VP		+123.6K	20.28	▲
SISTERHOOD	1	+1.1M	21.33	▲
	2	+935.5K	20.91	▲
	3	+433.1K	20.36	▲
	4	+407.0K	20.59	▲
PARENT	1	+1.1M	19.77	▼
	2	+9.0K	20.01	▼
	3	+2.9M	15.63	▼
COMP_IN		+17.4K	20.36	▲
REM_NPB		–3.5K	19.93	▼
REM_-C		–143.4K	19.3	▼
REM_SG		–9.4K	20.01	▼

Figure 3: For each relabeling method and variant, the impact on ruleset size and BLEU score over the baseline.

3 Relabeling

The small tagset of the PTB has the advantage of being simple to annotate and to parse. On the other hand, this can lead to tags that are overly generic. Klein and Manning (2003) discuss this as a problem in parsing and demonstrate that annotating additional information onto the PTB tags leads to improved parsing performance. We similarly propose methods of relabeling PTB trees that notably improve MT quality. In the next two subsections, we explore relabeling strategies that fall under two categories introduced by Klein and Manning — internal annotation and external annotation.

3.1 Internal Annotation

Internal annotation reveals information about a node and its descendants to its surrounding nodes (ancestors, sisters, and other relatives) that is otherwise hidden. This is paramount in MT because the contents of a node must be understood before the node can be reliably translated and positioned in a sentence. Here we discuss two such strategies: lexicalization and tag annotation.

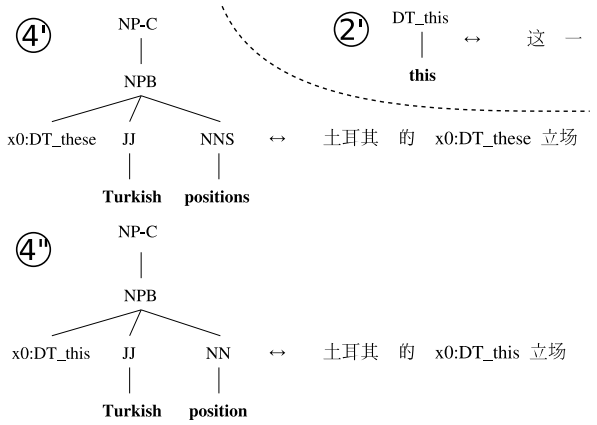

Figure 4: Rules before and after lexicalization.

3.1.1 Lexicalization

Many state-of-the-art statistical parsers incorporate *lexicalization* to effectively capture word-specific behavior, which has proved helpful in our system as well. We generalize lexicalization to allow a lexical item (terminal word) to be annotated onto any ancestor label, not only its parent.

Let us revisit the determiner/noun number disagreement problem in Figure 2 (**this Turkish positions*). If we lexicalize all **DT**s in the parse trees, the problematic **DT** is relabeled more specifically as **DT_this**, as seen in rule ②' in Figure 4. This also produces rules like ④', where both the determiner and the noun are plural (notice the **DT_these**), and ④'', where both are singular. With such a ruleset, ②' could only combine with ④'', not ④', enforcing the grammatical output *this Turkish position*.

We explored five lexicalization strategies, each targeting a different grammatical category. A common translation mistake was the improper choice of prepositions, e.g., *responsibility **to** attacks*. Lexicalizing prepositions proved to be the most effective lexicalization method (LEX_PREP). We annotated a preposition onto both its parent (**IN** or **TO**) and its grandparent (**PP**) since the generic **PP** tag was often at fault. We tried lexicalizing all prepositions (variant 1), the top 15 most common prepositions (variant 2), and the top 5 most common (variant 3). All gave statistically significant BLEU improvements, especially variant 2.

The second strategy was **DT** lexicalization

(LEX_DT), which we encountered previously in Figure 4. This addresses two features of Chinese that are problematic in translation to English: the infrequent use of articles and the lack of overt number indicators on nouns. We lexicalized these determiners: *the*, *a*, *an*, *this*, *that*, *these*, or *those*, and grouped together those with similar grammatical distributions (*a/an*, *this/that*, and *these/those*). Variant 1 included all the determiners mentioned above and variant 2 was restricted to *the* and *a/an* to focus only on articles. The second slightly improved on the first.

The third type was auxiliary lexicalization (LEX_AUX), in which all forms of the verb *be* are annotated with **_be**, and similarly with *do* and *have*. The PTB purposely eliminated such distinctions; here we seek to recover them. However, auxiliaries and verbs function very differently and thus cannot be treated identically. Klein and Manning (2003) make a similar proposal but omit *do*. Variants 1, 2, and 3, lexicalize *have*, *be*, and *do*, respectively. The third variant slightly outperformed the other variants, including variant 4, which combines all three.

The last two methods are drawn directly from Klein and Manning (2003). In **CC** lexicalization (LEX_CC), both *but* and *&* are lexicalized since these two conjunctions are distributed very differently compared to other conjunctions. Though helpful in parsing, it proved detrimental in our system. In **%** lexicalization (LEX_%), the percent sign (%) is given its own **PCT** tag rather than its typical **NN** tag, which gave a statistically significant BLEU increase.

3.1.2 Tag Annotation

In addition to propagating up a terminal word, we can also propagate up a nonterminal, which we call *tag annotation*. This partitions a grammatical category into more specific subcategories, but not as fine-grained as lexicalization. For example, a **VP** headed by a **VBG** can be tag-annotated as **VP_VBG** to represent a progressive verb phrase.

Let us once again return to Figure 2 to address the auxiliary/verb tense disagreement error (**has demonstrate*). The auxiliary *has* expects a **VP-C**, permitting the bare verb phrase *demonstrate* to be incorrectly used. However, if we tag-annotate all **VP-C**s, rule ⑥ would be relabeled as **VP-C_VB** in rule ⑥′ and rule ⑦ as ⑦′ in Figure 5. Rule ⑥′ can no longer

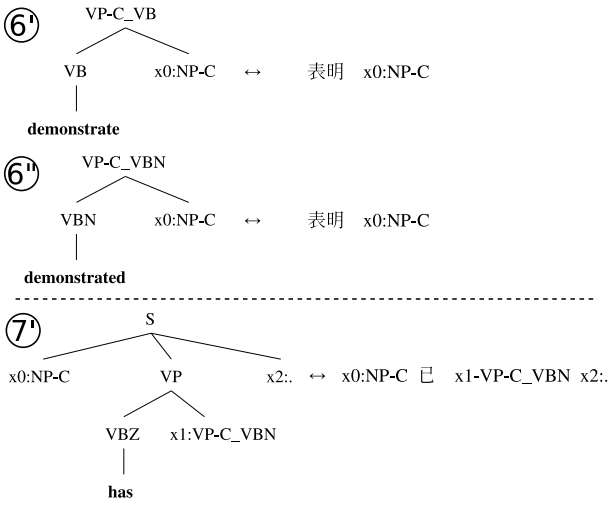

Figure 5: Rules before and after tag annotation.

join with ⑦′, while the variant rule ⑥′ can, which produces the grammatical result *has demonstrated*.

We noticed many wrong verb tense choices, e.g., gerunds and participles used as main sentence verbs. We resolved this by tag-annotating every **VP** and **VP-C** with its head verb (TAG_VP). Note that we group **VBZ** and **VBP** together since they have very similar grammatical distributions and differ only by number. This strategy gave a healthy BLEU improvement.

3.2 External Annotation

In addition to passing information from inside a node to the outside, we can pass information from the external environment into the node through *external annotation*. This allows us to make translation decisions based on the context in which a word or phrase is found. In this subsection, we look at three such methods: sisterhood annotation, parent annotation, and complement annotation.

3.2.1 Sisterhood Annotation

The single most effective relabeling scheme we tried was *sisterhood annotation*. We annotate each nonterminal with **#L** if it has any sisters to the left, **#R** if any to the right, **#LR** if on both sides, and nothing if it has no sisters. This distinguishes between words that tend to fall on the left or right border of a constituent (often head words, like **NN#L** in an **NP** or **IN#R** in a **PP**), in the middle of a constituent (often modifiers, like **JJ#LR** in an **NP**), or by themselves

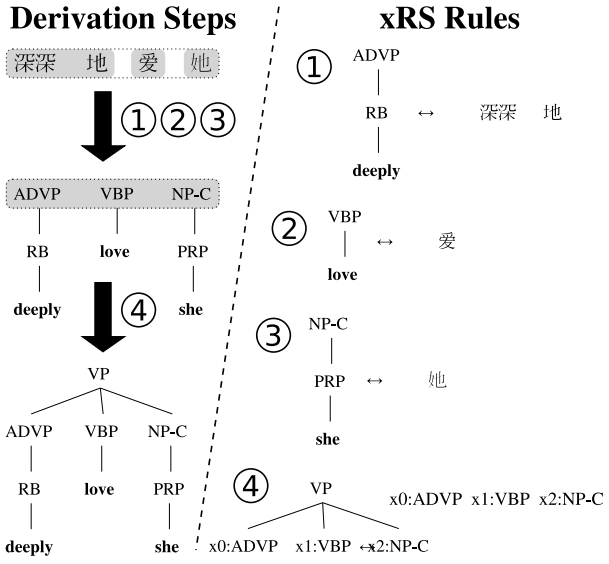

Figure 6: A bad translation fixable by sisterhood or parent annotation.

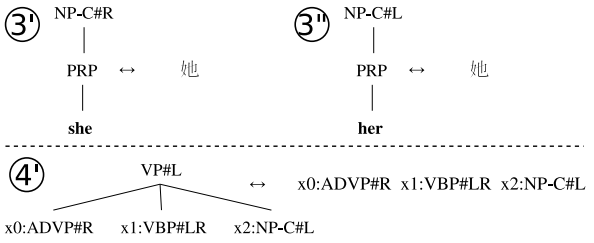

Figure 7: Rules before and after sisterhood annotation.

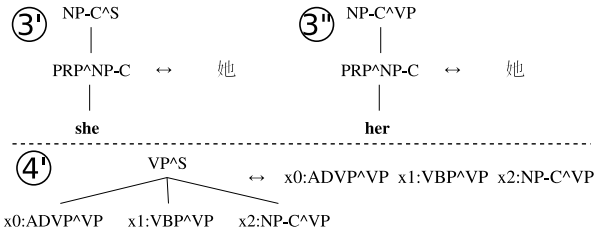

Figure 8: Rules before and after parent annotation.

(often particles and pronouns, like **RP** and **PRP**). In our outputs, we frequently find words used in positions where they should be disallowed or disfavored.

Figure 6 presents a derivation that leads to the ungrammatical output **deeply love she*. The subject pronoun *she* is incorrectly preferred over the object form *her* because the most popular **NP-C** translation for 她 is *she*. We can sidestep this mistake through sisterhood-annotation, which yields the relabeled rules ③' and ④' in Figure 7. Rule ④' expects an **NP-C** on the right border of the constituent (**NP-C#L**). Since *she* never occurs in this position in the PTB, it should never be sisterhood-annotated as an **NP-C#L**. It does occur with sisters to the right, which gives the **NP-C#R** rule ③'. The object **NP-C** *her*, on the other hand, is frequently rightmost in a constituent, which is reflected in the **NP-C#L** rule ③''. Using this rule with rule ④' gives the desired result *deeply love her*.

We experimented with four sisterhood annotation (SISTERHOOD) variants of decreasing complexity. The first was described above, which includes rightmost (**#L**), leftmost (**#R**), middle (**#LR**), and alone (no annotation). Variant 2 omitted **#LR**, variant 3 kept only **#LR**, and variant 4 only annotated nodes without sisters. Variants 1 and 2 produced the largest gains from relabeling: 1.27 and 0.85 BLEU points, respectively.

3.2.2 Parent Annotation

Another common relabeling method in parsing is *parent annotation* (Johnson, 1998), in which a node is annotated with its parent's label. Typically, this is done only to nonterminals, but Klein and Manning (2003) found that annotating preterminals as well was highly effective. It seemed likely that such contextual information could also benefit MT.

Let us tackle the bad output from Figure 6 with parent annotation. In Figure 8, rule ④ is relabeled as rule ④' and expects an **NP-CˆVP**, i.e., an **NP-C** with a **VP** parent. In the PTB, we observe that the **NP-C** *she* never has a **VP** parent, while *her* does. In fact, the most popular parent for the **NP-C** *her* is **VP**, while the most popular parent for *she* is **S**. Rule ③ is relabeled as the **NP-CˆS** rule ③' and *her* is expressed as the **NP-CˆVP** rule ③''. Only rule ③'' can partner with rule ④', which produces the correct output *deeply love her*.

We tested three variants of parent annotation (PARENT): (1) all nonterminals are parent-annotated, (2) only **S** nodes are parent-annotated, and (3) all nonterminals are parent- and grandparent-annotated (the annotation of a node's parent's parent). The first and third variants yielded the largest ruleset sizes of all relabeling methods. The second variant was restricted only to **S** to capture the difference between top-level clauses (**SˆTOP**) and em-

bedded clauses (like **S^S-C**). Unfortunately, all three variants turned out to be harmful in terms of BLEU.

3.2.3 Complement Annotation

In addition to a node's parent, we can also annotate a node's complement. This captures the fact that words have a preference of taking certain complements over others. For instance, 96% of cases where the **IN** *of* takes one complement in the PTB, it takes **NP-C**. On the other hand, *although* never takes **NP-C** but takes **S-C** 99% of the time.

Consider the derivation in Figure 9 that results in the bad output **postponed out May 6*. The **IN** *out* is incorrectly allowed despite the fact that it almost never takes an **NP-C** complement (0.6% of cases in the PTB). A way to restrict this is to annotate the **IN**'s complement. Complement-annotated versions of rules ② and ③ are given in Figure 10. Rule ② is relabeled as the **IN/PP-C** rule ②' since **PP-C** is the most common complement for *out* (99% of the time). Since rule ③' expects an **IN/NP-C**, rule ②' is disqualified. The preposition *from* (rule ②''), on the other hand, frequently takes **NP-C** as complement (82% of the time). Combining rule ②'' with rule ③' ensures the correct output *postponed from May 6*.

Complement-annotating all **IN** tags with their complement if they had one and only one complement (COMP_IN) gave a significant BLEU improvement with only a modest increase in ruleset size.

3.3 Removal of Parser Annotations

Many parsers, though trained on the PTB, do not preserve the original tagset. They may omit function tags (like **-TMP**), indices, and null/gap elements or add annotations to increase parsing accuracy and provide useful grammatical information. It is not obvious whether these modifications are helpful for MT, so we explore the effects of removing them.

The statistical parser we used makes three relabelings: (1) base **NP**s are relabeled as **NPB**, (2) argument nonterminals are suffixed with **-C**, and (3) subjectless sentences are relabeled from **S** to **SG**. We tried removing each annotation individually (REM_NPB, REM_-C, and REM_SG), but doing so significantly dropped the BLEU score. This leads us to conclude these parser additions are helpful in MT.

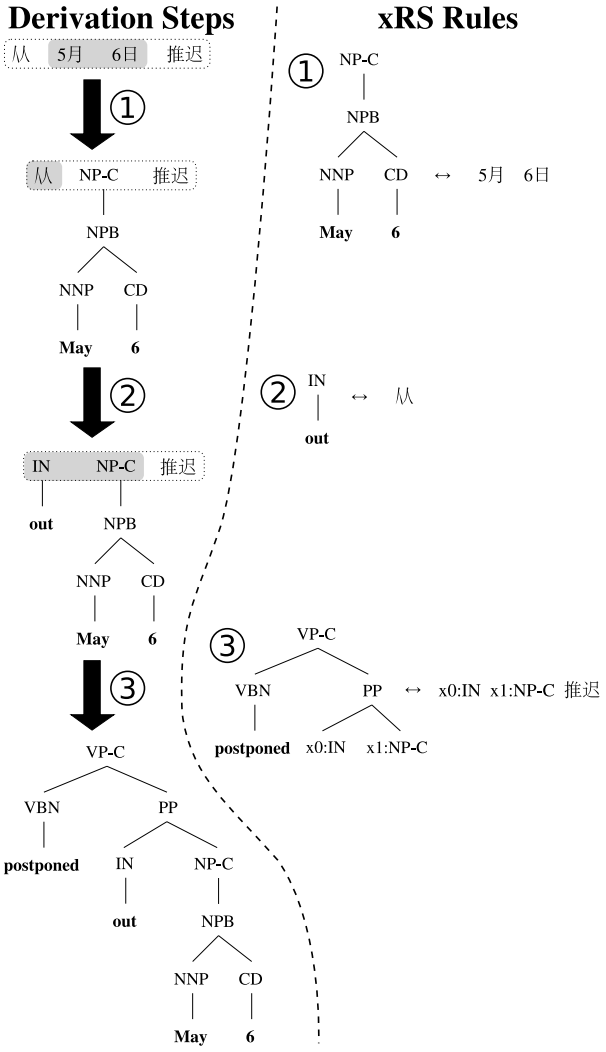

Figure 9: A bad translation fixable by complement annotation.

Figure 10: Rules before and after complement annotation.

4 Evaluation

To maximize the benefit of relabeling, we incorporated five of the most promising relabeling strategies into one additive system: LEX_%, LEX_DT variant

		Δ # Rules		BLEU	
Relabeling	**Variant**	**Ind.**	**Cum.**	**Ind.**	**Cum.**
BASELINE		—	—	20.06	20.06
LEX_%		+0.3K	+0.3K	20.14	20.14
LEX_DT	2	+29.6K	+29.9K	20.18	20.3
TAG_VP		+123.6K	+153.5K	20.28	20.43
LEX_PREP	2	+254.8K	+459.0K	20.36	21.25
SISTERHOOD	1	+1.1M	+1.5M	21.33	*22.38*

Figure 11: Relabelings in the additive system and their individual/cumulative effects over the baseline.

2, TAG_VP, LEX_PREP variant 2, and SISTERHOOD variant 1. These relabelings contributed to a 2.3 absolute (11.6% relative) BLEU point increase over the baseline, with a score of 22.38. Figure 11 lists these relabelings in the order they were added.

5 Conclusion

We have demonstrated that relabeling syntax trees for use in syntax-based machine translation can significantly boost translation performance. It is naïve to assume that linguistic resources can be immediately useful out of the box, in our case, the Penn Treebank for MT. Rather, we targeted features of the PTB tagset that impair translatability and proposed relabeling strategies to overcome these weaknesses. Many of our ideas effectively raised the BLEU score over a baseline system without relabeling. Finally, we demonstrated through an additive system that relabelings can be combined together to achieve an even greater improvement in translation quality.

Acknowledgments

This research was supported in part by NSF grant IIS-0428020. We would like to thank Greg Langmead, Daniel Marcu, and Wei Wang for helpful comments. This paper describes work conducted while the first author was at the University of Southern California/Information Sciences Institute.

References

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 2000. Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, 26(1):45–60.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL-05*.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL-05*, pages 531–540.

Michael Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of ACL-03 (Companion Volume)*, pages 205–208.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of HLT/NAACL-04*, pages 273–280.

Dan Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of ACL-03*.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *Proceedings of HLT/NAACL-04*, pages 105–112.

Mary Hearne and Andy Way. 2003. Seeing the wood for the trees: Data-Oriented Translation. In *Proceedings of MT Summit IX*.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL-03*, pages 423–430.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL-03*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP-04*.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP-02*.

Mitchell Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of ACL-04*, pages 653–660.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL-00*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL-02*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal SMT. In *Proceedings of ACL-05*, pages 271–279.

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

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING-04*.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL-01*.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: how much improvement do we need to have a better system? In *Proceedings of LREC-04*.