The impact of parse quality on syntactically-informed statistical machine translation

Chris Quirk and **Simon Corston-Oliver**
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
{chrisq,simonco}@microsoft.com

Abstract

We investigate the impact of parse quality on a syntactically-informed statistical machine translation system applied to technical text. We vary parse quality by varying the amount of data used to train the parser. As the amount of data increases, parse quality improves, leading to improvements in machine translation output and results that significantly outperform a state-of-the-art phrasal baseline.

1 Introduction

The current study is a response to a question that proponents of syntactically-informed machine translation frequently encounter: How sensitive is a syntactically-informed machine translation system to the quality of the input syntactic analysis? It has been shown that phrasal machine translation systems are not affected by the quality of the input word alignments (Koehn et al., 2003). This finding has generally been cast in favorable terms: such systems are robust to poor quality word alignment. A less favorable interpretation of these results might be to conclude that phrasal statistical machine translation (SMT) systems do not stand to benefit from improvements in word alignment.

In a similar vein, one might ask whether contemporary syntactically-informed machine translation systems would benefit from improvements in parse accuracy. One possibility is that current syntactically-informed SMT systems are deriving only limited value from the syntactic analyses, and would therefore not benefit from improved analyses. Another possibility is that syntactic analysis does indeed contain valuable information that could be exploited by machine learning techniques, but that current parsers are not of sufficient quality to be of use in SMT.

With these questions and concerns, let us begin. Following some background discussion we describe a set of experiments intended to elucidate the impact of parse quality on SMT.

2 Background

We trained statistical machine translation systems on technical text. In the following sections we provide background on the data used for training, the dependency parsing framework used to produce treelets, the treelet translation framework and salient characteristics of the target languages.

2.1 Dependency parsing

Dependency analysis is an alternative to constituency analysis (Tesnière, 1959; Melčuk, 1988). In a dependency analysis of syntax, words directly modify other words, with no intervening non-lexical nodes. We use the terms child node and parent node to denote the tokens in a dependency relation. Each child has a single parent, with the lexical root of the sentence dependent on a synthetic ROOT node.

We use the parsing approach described in (Corston-Oliver et al., 2006). The parser is trained on dependencies extracted from the English Penn Treebank version 3.0 (Marcus et al., 1993) by using the head-percolation rules of (Yamada and Matsumoto, 2003).

Given a sentence x, the goal of the parser is to find the highest-scoring parse $\hat{y}$ among all possible parses $y \in Y$:

$$\hat{y} = \arg\max_{y \in Y} s(x, y) \qquad (1)$$

The score of a given parse y is the sum of the

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 62–69,
Sydney, July 2006. ©2006 Association for Computational Linguistics

scores of all its dependency links $(i,j) \in y$:

$$s(x,y) = \sum_{(i,j)\in y} d(i,j) = \sum_{(i,j)\in y} \mathbf{w}\cdot\mathbf{f}(i,j) \quad (2)$$

where the link (i,j) indicates a parent-child dependency between the token at position i and the token at position j. The score $d(i,j)$ of each dependency link (i,j) is further decomposed as the weighted sum of its features $\mathbf{f}(i,j)$.

The feature vector $\mathbf{f}(i,j)$ computed for each possible parent-child dependency includes the part-of-speech (POS), lexeme and stem of the parent and child tokens, the POS of tokens adjacent to the child and parent, and the POS of each token that intervenes between the parent and child. Various combinations of these features are used, for example a new feature is created that combines the POS of the parent, lexeme of the parent, POS of the child and lexeme of the child. Each feature is also conjoined with the direction and distance of the parent, e.g. does the child precede or follow the parent, and how many tokens intervene?

To set the weight vector $\mathbf{w}$, we train twenty averaged perceptrons (Collins, 2002) on different shuffles of data drawn from sections 02–21 of the Penn Treebank. The averaged perceptrons are then combined to form a Bayes Point Machine (Herbrich et al., 2001; Harrington et al., 2003), resulting in a linear classifier that is competitive with wide margin techniques.

To find the optimal parse given the weight vector $\mathbf{w}$ and feature vector $\mathbf{f}(i,j)$ we use the decoder described in (Eisner, 1996).

2.2 Treelet translation

For syntactically-informed translation, we follow the treelet translation approach described in (Quirk et al., 2005). In this approach, translation is guided by treelet translation pairs. Here, a *treelet* is a connected subgraph of a dependency tree. A treelet translation pair consists of a source treelet S, a target treelet T, and a word alignment $A \subset S \times T$ such that for all $s \in S$, there exists a unique $t \in T$ such that $(s,t) \in A$, and if t is the root of T, there is a unique $s \in S$ such that $(s,t) \in A$.

Translation of a sentence begins by parsing that sentence into a dependency representation. This dependency graph is partitioned into treelets; like (Koehn et al., 2003), we assume a uniform probability distribution over all partitions. Each source treelet is matched to a treelet translation pair; together, the target language treelets in those treelet translation pairs will form the target translation. Next the target language treelets are joined to form a single tree: the parent of the root of each treelet is dictated by the source. Let t_r be the root of the target language treelet, and s_r be the source node aligned to it. If s_r is the root of the source sentence, then t_r is made the root of the target language tree. Otherwise let s_p be the parent of s_r, and t_p be the target node aligned to s_p: t_r is attached to t_p. Finally the ordering of all the nodes is determined, and the target tree is specified, and the target sentence is produced by reading off the labels of the nodes in order.

Translations are scored according to a log-linear combination of feature functions, each scoring different aspects of the translation process. We use a beam search decoder to find the best translation T^* according to the log-linear combination of models:

$$T^* = \arg\max_T \left\{ \sum_{f\in F} \lambda_f f(S,T,A) \right\} \quad (3)$$

The models include inverted and direct channel models estimated by relative frequency, lexical weighting channel models following (Vogel et al., 2003), a trigram target language model using modified Kneser-Ney smoothing (Goodman, 2001), an order model following (Quirk et al., 2005), and word count and phrase count functions. The weights for these models are determined using the method described in (Och, 2003).

To estimate the models and extract the treelets, we begin from a parallel corpus. First the corpus is word-aligned using GIZA++ (Och and Ney, 2000), then the source sentence are parsed, and finally dependencies are projected onto the target side following the heuristics described in (Quirk et al., 2005). This word aligned parallel dependency tree corpus provides training material for an order model and a target language tree-based language model. We also extract treelet translation pairs from this parallel corpus. To limit the combinatorial explosion of treelets, we only gather treelets that contain at most four words and at most two gaps in the surface string. This limits the number of mappings to be $O(n^3)$ in the worst case, where n is the number of nodes in the dependency tree.

2.3 Language pairs

In the present paper we focus on English-to-German and English-to-Japanese machine transla-

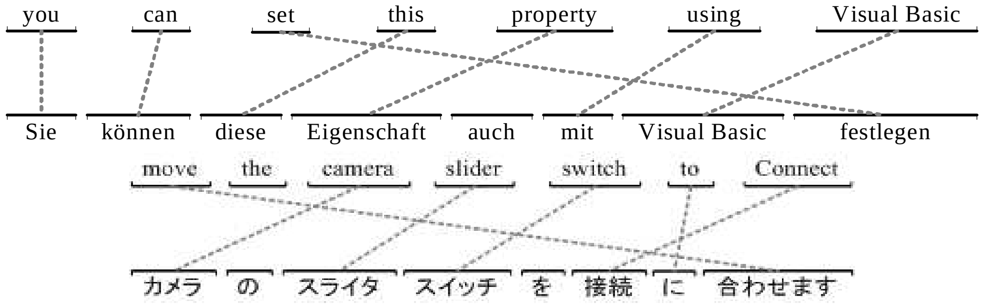


Figure 1: Example German-English and Japanese-English sentence pairs, with word alignments.

tion. Both German and Japanese differ markedly from English in ways that we believe illuminate well the strengths of a syntactically-informed SMT system. We provide a brief sketch of the linguistic characteristics of German and Japanese relevant to the present study.

2.3.1 German

Although English and German are closely related – they both belong to the western branch of the Germanic family of Indo-European languages – the languages differ typologically in ways that are especially problematic for current approaches to statistical machine translation as we shall now illustrate. We believe that these typological differences make English-to-German machine translation a fertile test bed for syntax-based SMT.

German has richer inflectional morphology than English, with obligatory marking of case, number and lexical gender on nominal elements and person, number, tense and mood on verbal elements. This morphological complexity, combined with pervasive, productive noun compounding is problematic for current approaches to word alignment (Corston-Oliver and Gamon, 2004).

Equally problematic for machine translation is the issue of word order. The position of verbs is strongly determined by clause type. For example, in main clauses in declarative sentences, finite verbs occur as the second constituent of the sentence, but certain non-finite verb forms occur in final position. In Figure 1, for example, the English “can” aligns with German “können” in second position and “set” aligns with German “festlegen” in final position.

Aside from verbs, German is usually characterized as a “free word-order” language: major constituents of the sentence may occur in various orders, so-called “separable prefixes” may occur bound to the verb or may detach and occur at a considerable distance from the verb on which they depend, and extraposition of various kinds of subordinate clause is common. In the case of extraposition, for example, more than one third of relative clauses in human-translated German technical text are extraposed. For comparable English text the figure is considerably less than one percent (Gamon et al., 2002).

2.3.2 Japanese

Word order in Japanese is rather different from English. English has the canonical constituent order subject-verb-object, whereas Japanese prefers subject-object-verb order. Prepositional phrases in English generally correspond to postpositional phrases in Japanese. Japanese noun phrases are strictly head-final whereas English noun phrases allow postmodifiers such as prepositional phrases, relative clauses and adjectives. Japanese has little nominal morphology and does not obligatorily mark number, gender or definiteness. Verbal morphology in Japanese is complex with morphological marking of tense, mood, and politeness. Topicalization and subjectless clauses are pervasive, and problematic for current SMT approaches.

The Japanese sentence in Figure 1 illustrates several of these typological differences. The sentence-initial imperative verb “move” in the English corresponds to a sentence-final verb in the Japanese. The Japanese translation of the object noun phrase “the camera slider switch” precedes the verb in Japanese. The English preposition “to” aligns to a postposition in Japanese.

3 Experiments

Our goal in the current paper is to measure the impact of parse quality on syntactically-informed statistical machine translation. One method for producing parsers of varying quality might be to train a parser and then to transform its output, e.g.

by replacing the parser's selection of the parent for certain tokens with different nodes.

Rather than randomly adding noise to the parses, we decided to vary the quality in ways that more closely mimic the situation that confronts us as we develop machine translation systems. Annotating data for POS requires considerably less human time and expertise than annotating syntactic relations. We therefore used an automatic POS tagger (Toutanova et al., 2003) trained on the complete training section of the Penn Treebank (sections 02–21). Annotating syntactic dependencies is time consuming and requires considerable linguistic expertise.[1] We can well imagine annotating syntactic dependencies in order to develop a machine translation system by annotating first a small quantity of data, training a parser, training a system that uses the parses produced by that parser and assessing the quality of the machine translation output. Having assessed the quality of the output, one might annotate additional data and train systems until it appears that the quality of the machine translation output is no longer improving. We therefore produced parsers of varying quality by training on the first *n* sentences of sections 02–21 of the Penn Treebank, where *n* ranged from 250 to 39,892 (the complete training section). At training time, the gold-standard POS tags were used. For parser evaluation and for the machine translation experiments reported here, we used an automatic POS tagger (Toutanova et al., 2003) trained on sections 02–21 of the Penn Treebank.

We trained English-to-German and English-to-Japanese treelet translation systems on approximately 500,000 manually aligned sentence pairs drawn from technical computer documentation. The sentence pairs consisted of the English source sentence and a human-translation of that sentence. Table 1 summarizes the characteristics of this data. Note that German vocabulary and singleton counts are slightly more than double the corresponding English counts due to complex morphology and pervasive compounding (see section 2.3.1).

3.1 Parser accuracy

To evaluate the accuracy of the parsers trained on different samples of sentences we used the traditional blind test section of the Penn Treebank (section 23). As is well-known in the parsing community, parse quality degrades when a parser trained on the Wall Street Journal text in the Penn Treebank is applied to a different genre or semantic domain. Since the technical materials that we were training the translation system on differ from the Wall Street Journal in lexicon and syntax, we annotated a set of 250 sentences of technical material to use in evaluating the parser. Each of the authors independently annotated the same set of 250 sentences. The annotation took less than six hours for each author to complete. Inter-annotator agreement excluding punctuation was 91.8%. Differences in annotation were resolved by discussion, and the resulting set of annotations was used to evaluate the parsers.

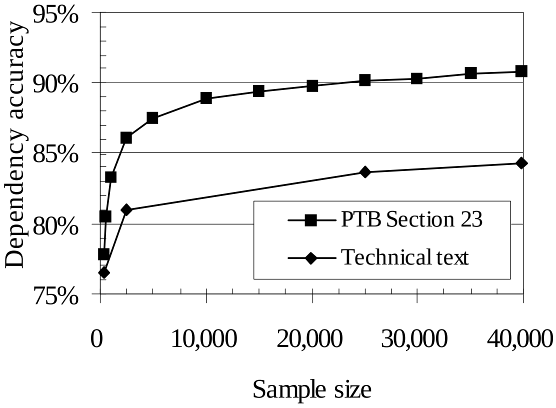


Figure 2: Unlabeled dependency accuracy of parsers trained on different numbers of sentences. The graph compares accuracy on the blind test section of the Penn Treebank to accuracy on a set of 250 sentences drawn from technical text. Punctuation tokens are excluded from the measurement of dependency accuracy.

Figure 2 shows the accuracy of parsers trained on samples of various sizes, excluding punctuation tokens from the evaluation, as is customary in evaluating dependency parsers. When measured against section 23 of the Penn Treebank, the section traditionally used for blind evaluation, the parsers range in accuracy from 77.8% when trained on 250 sentences to 90.8% when trained on all of sections 02–21. As expected, parse accuracy degrades when measured on text that differs greatly from the training text. A parser trained on 250 Penn Treebank sentences has a dependency

[1] Various people have suggested to us that the linguistic expertise required to annotate syntactic dependencies is less than the expertise required to apply a formal theory of constituency like the one that informs the Penn Treebank. We tend to agree, but have not put this claim to the test.

		English	German	English	Japanese
Training	Sentences	515,318		500,000	
	Words	7,292,903	8,112,831	7,909,198	9,379,240
	Vocabulary	59,473	134,829	66,731	68,048
	Singletons	30,452	66,724	50,381	52,911
Test	Sentences	2,000		2,000	
	Words	28,845	31,996	30,616	45,744

Table 1: Parallel data characteristics

accuracy of 76.6% on the technical text. A parser trained on the complete Penn Treebank training section has a dependency accuracy of 84.3% on the technical text.

Since the parsers make extensive use of lexical features, it is not surprising that the performance on the two corpora should be so similar with only 250 training sentences; there were not sufficient instances of each lexical item to train reliable weights or lexical features. As the amount of training data increases, the parsers are able to learn interesting facts about specific lexical items, leading to improved accuracy on the Penn Treebank. Many of the lexical items that occur in the Penn Treebank, however, occur infrequently or not at all in the technical materials so the lexical information is of little benefit. This reflects the mismatch of content. The Wall Street Journal articles in the Penn Treebank concern such topics as world affairs and the policies of the Reagan administration; these topics are absent in the technical materials. Conversely, the Wall Street Journal articles contain no discussion of such topics as the intricacies of SQL database queries.

3.2 Translation quality

Table 2 presents the impact of parse quality on a treelet translation system, measured using BLEU (Papineni et al., 2002). Since our main goal is to investigate the impact of parser accuracy on translation quality, we have varied the parser training data, but have held the MT training data, part-of-speech-tagger, and all other factors constant. We observe an upward trend in BLEU score as more training data is made available to the parser; the trend is even clearer in Japanese.[2] As a baseline, we include right-branching dependency trees, i.e., trees in which the parent of each word is its left neighbor and the root of a sentence is the first word. With this analysis, treelets are simply subsequences of the sentence, and therefore are very similar to the phrases of Phrasal SMT. In English-to-German, this result produces results very comparable to a phrasal SMT system (Koehn et al., 2003) trained on the same data. For English-to-Japanese, however, this baseline performs much worse than a phrasal SMT system. Although phrases and treelets should be nearly identical under this scenario, the decoding constraints are somewhat different: the treelet decoder assumes phrasal cohesion during translation. This constraint may account for the drop in quality.

	EG	EJ
Phrasal decoder	31.7±1.2	32.9±0.9
Treelet decoder		
Right-branching	31.4±1.3	28.0±0.7
250 sentences	32.8±1.4	34.1±0.9
2,500 sentences	33.0±1.4	34.6±1.0
25,000 sentences	33.7±1.5	35.7±0.9
39,892 sentences	33.6±1.5	36.0±1.0

Table 2: BLEU score vs. decoder and parser variants. Here sentences refer to the amount of parser training data, not MT training data.

Since the confidence intervals for many pairs overlap, we ran pairwise tests for each system to determine which differences were significant at the $p < 0.05$ level using the bootstrap method described in (Zhang and Vogel, 2004); Table 3 summarizes this comparison. Neither language pair achieves a statistically significant improvement from increasing the training data from 25,000 pairs to the full training set; this is not surprising since the increase in parse accuracy is quite small (90.2% to 90.8% on Wall Street Journal text).

To further understand what differences in dependency analysis were affecting translation quality, we compared a treelet translation system that

[2] This is particularly encouraging since various people have remarked to us that syntax-based SMT systems may be disadvantaged under n-gram scoring techniques such as BLEU.

	Pharaoh	Right-branching	250	2,500	25,000	39,892
Pharaoh		$\sim$	$>$	$>$	$>$	$>$
Right-branching			$>$	$>$	$>$	$>$
250				$\sim$	$>$	$>$
2,500					$>$	$>$
25,000						$\sim$

(a) English-German

	Pharaoh	Right-branching	250	2,500	25,000	39,892
Pharaoh		$<$	$\sim$	$>$	$>$	$>$
Right-branching			$>$	$>$	$>$	$>$
250				$>$	$>$	$>$
2,500					$>$	$>$
25,000						$\sim$

(b) English-Japanese

Table 3: Pairwise statistical significance tests. $>$ indicates that the system on the top is significantly better than the system on the left; $<$ indicates that the system on top is significantly worse than the system on the left; $\sim$ indicates that difference between the two systems is not statistically significant.

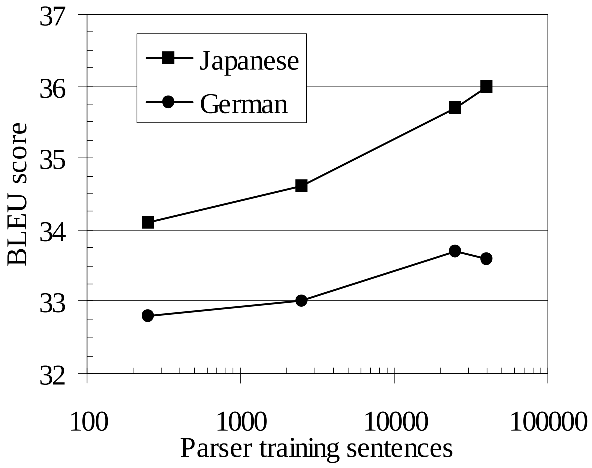


Figure 3: BLEU score vs. number of sentences used to train the dependency parser

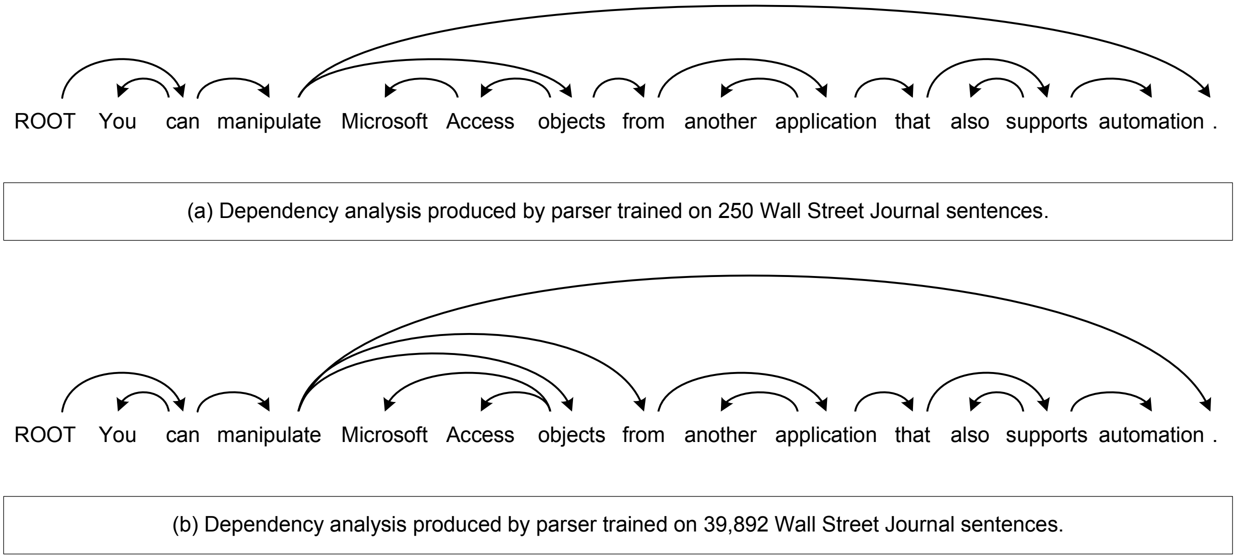
used a parser trained on 250 Penn Treebank sentences to a treelet translation system that used a parser trained on 39,892 Treebank sentences. From the test data, we selected 250 sentences where these two parsers produced different analyses. A native speaker of German categorized the differences in machine translation output as either improvements or regressions. We then examined and categorized the differences in the dependency analyses. Table 4 summarizes the results of this comparison. Note that this table simply identifies correlations between parse changes and translation changes; it does not attempt to identify a causal link. In the analysis, we borrow the term "NP [Noun Phrase] identification" from constituency analysis to describe the identification of dependency treelets spanning complete noun phrases.

There were 141 sentences for which the machine translated output improved, 71 sentences for which the output regressed and 38 sentences for which the output was identical. Improvements in the attachment of prepositions, adverbs, gerunds and dependent verbs were common amongst improved translations, but rare amongst regressed translations. Correct identification of the dependent of a preposition[3] was also much more common amongst improvements.

Certain changes, such as improved root identification and final punctuation attachment, were very common across the corpus. Therefore their common occurrence amongst regressions is not very surprising. It was often the case that improvements in root identification or final punctuation attachment were offset by regressions elsewhere in the same sentence.

Improvements in the parsers are cases where the syntactic analysis more closely resembles the analysis of dependency structure that results from applying Yamada and Matsumoto's head-finding rules to the Penn Treebank. Figure 4 shows different parses produced by parsers trained on dif-

[3] In terms of constituency analysis, a prepositional phrase should consist of a preposition governing a single noun phrase


ROOT You can manipulate Microsoft Access objects from another application that also supports automation .
ROOT You can manipulate Microsoft Access objects from another application that also supports automation .

(a) Dependency analysis produced by parser trained on 250 Wall Street Journal sentences.

(b) Dependency analysis produced by parser trained on 39,892 Wall Street Journal sentences.

Figure 4: Parses produced by parsers trained on different numbers of sentences.

ferent numbers of sentences. The parser trained on 250 sentences incorrectly attaches the preposition "from" as a dependent of the noun "objects" whereas the parser trained on the complete Penn Treebank training section correctly attaches the preposition as a dependent of the verb "manipulate". These two parsers also yield different analyses of the phrase "Microsoft Access objects". In parse (a), "objects" governs "Office" and "Office" in turn governs "Microsoft". This analysis is linguistically well-motivated, and makes a treelet spanning "Microsoft Office" available to the treelet translation system. In parse (b), the parser has analyzed this phrase so that "objects" directly governs "Microsoft" and "Office". The analysis more closely reflects the flat branching structure of the Penn Treebank but obscures the affinity of "Microsoft" and "Office".

An additional measure of parse utility for MT is the amount of translation material that can be extracted from a parallel corpus. We increased the parser training data from 250 sentences to 39,986 sentences, but held the number of aligned sentence pairs used train other modules constant. The count of treelet translation pairs occurring at least twice in the English-German parallel corpus grew from 1,895,007 to 2,010,451.

4 Conclusions

We return now to the questions and concerns raised in the introduction. First, is a treelet SMT system sensitive to parse quality? We have shown that such a system *is* sensitive to the quality of the input syntactic analyses. With the less accurate parsers that result from training on extremely small numbers of sentences, performance is comparable to state-of-the-art phrasal SMT systems. As the amount of data used to train the parser increases, both English-to-German and English-to-Japanese treelet SMT improve, and produce results that are statistically significantly better than the phrasal baseline.

Error category	Regress	Improve
Attachment of prep	1%	22%
Root identification	13%	28%
Final punctuation	18%	30%
Coordination	6%	16%
Dependent verbs	14%	32%
Arguments of verb	6%	15%
NP identification	24%	33%
Dependent of prep	0%	7%
Other attachment	3%	22%

Table 4: Error analysis, showing percentage of regressed and improved translations exhibiting a parse improvement in each specified category

In the introduction we mentioned the concern that others have raised when we have presented our research: syntax might contain valuable information but current parsers might not be of sufficient quality. It is certainly true that the accuracy of the best parser used here falls well short of what we might hope for. A parser that achieves 90.8% dependency accuracy when trained on the Penn Treebank Wall Street Journal corpus and evaluated on comparable text degrades to 84.3% accuracy when evaluated on technical text. Despite the degradation in parse accuracy caused by the dramatic differences between the Wall Street Journal text and the technical articles, the treelet SMT system was able to extract useful patterns. Research on syntactically-informed SMT is not impeded by the accuracy of contemporary parsers.

One significant finding is that as few as 250 sentences suffice to train a dependency parser for use in the treelet SMT framework. To date our research has focused on translation from English to other languages. One concern in applying the treelet SMT framework to translation from languages other than English has been the expense of data annotation: would we require 40,000 sentences annotated for syntactic dependencies, i.e., an amount comparable to the Penn Treebank, in order to train a parser that was sufficiently accurate to achieve the machine translation quality that we have seen when translating from English? The current study gives hope that source languages can be added with relatively modest investments in data annotation. As more data is annotated with syntactic dependencies and more accurate parsers are trained, we would hope to see similar improvements in machine translation output.

We challenge others who are conducting research on syntactically-informed SMT to verify whether or to what extent their systems are sensitive to parse quality.

References

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

Simon Corston-Oliver and Michael Gamon. 2004. Normalizing German and English inflectional morphology to improve statistical word alignment. In R. E. Frederking and K. B. Taylor, editors, *Machine translation: From real users to research*. Springer Verlag.

Simon Corston-Oliver, Anthony Aue, Kevin Duh, and Eric Ringger. 2006. Multilingual dependency parsing using Bayes Point Machines. In *Proceedings of HLT/NAACL*.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of COLING*, pages 340–345.

Michael Gamon, Eric Ringger, Zhu Zhang, Robert Moore, and Simon Corston-Oliver. 2002. Extraposition: A case study in German sentence realization. In *Proceedings of COLING*, pages 301–307.

Joshua Goodman. 2001. A bit of progress in language modeling, extended version. Technical Report MSR-TR-2001-72, Microsoft Research.

Edward Harrington, Ralf Herbrich, Jyrki Kivinen, John C. Platt, and Robert C. Williamson. 2003. Online bayes point machines. In *Proc. 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 241–252.

Ralf Herbrich, Thore Graepel, and Colin Campbell. 2001. Bayes Point Machines. *Journal of Machine Learning Research*, pages 245–278.

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

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Igor A. Melčuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the ACL*, pages 440–447, Hongkong, China, October.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318, Philadelpha, Pennsylvania.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the ACL*.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Librairie C. Klincksieck.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT/EMNLP*, pages 252–259.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venugopal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical machine translation system. In *Proceedings of the MT Summit*.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, pages 195–206.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for mt evaluation metrics. In *Proceedings of TMI*.