Grammar Comparison Study for Translational Equivalence Modeling and Statistical Machine Translation

Min Zhang[1], Hongfei Jiang[2], Haizhou Li[1], Aiti Aw[1] and Sheng Li[2]
[1]Institute for Infocomm Research, Singapore
[2]Harbin Institute of Technology, China
{mzhang, hli, aaiti}@i2r.a-star.edu.sg
{hfjiang, lisheng}@mtlab.hit.edu.cn

Abstract

This paper presents a general platform, namely synchronous tree sequence substitution grammar (STSSG), for the grammar comparison study in Translational Equivalence Modeling (TEM) and Statistical Machine Translation (SMT). Under the STSSG platform, we compare the expressive abilities of various grammars through synchronous parsing and a real translation platform on a variety of Chinese-English bilingual corpora. Experimental results show that the STSSG is able to better explain the data in parallel corpora than other grammars. Our study further finds that the complexity of structure divergence is much higher than suggested in literature, which imposes a big challenge to syntactic transformation-based SMT.

1 Introduction

Translational equivalence is a mathematical relation that holds between linguistic expressions with the same meaning (Wellington et al., 2006). The common explicit representations of this relation are word alignments, phrase alignments and structure alignments between bilingual sentences. Translational Equivalence Modeling (TEM) is a process to describe and build these alignments using mathematical models. Thus, the study of TEM is highly relevant to Statistical Machine Translation (SMT).

Grammar is the most important infrastructure for TEM and SMT since translation models' expressive and generative abilities are mainly determined by the grammar. Many grammars, such as finite-state grammars (FSG), bracket/inversion transduction grammars (BTG/ITG) (Wu, 1997), context-free grammar (CFG), tree substitution grammar (TSG) (Comon et al., 2007) and their synchronous versions, have been explored in SMT. Based on these grammars, a great number of SMT models have been recently proposed, including string-to-string model (Synchronous FSG) (Brown et al., 1993; Koehn et al., 2003), tree-to-string model (TSG-string) (Huang et al., 2006; Liu et al., 2006; Liu et al., 2007), string-to-tree model (string-CFG/TSG) (Yamada and Knight, 2001; Galley et al., 2006; Marcu et al., 2006), tree-to-tree model (Synchronous CFG/TSG, Data-Oriented Translation) (Chiang, 2005; Cowan et al., 2006; Eisner, 2003; Ding and Palmer, 2005; Zhang et al., 2007; Bod, 2007; Quirk wt al., 2005; Poutsma, 2000; Hearne and Way, 2003) and so on.

Although many achievements have been obtained by these advances, it is still unclear which of these important pursuits is able to best explain human translation data, as each has its advantages and disadvantages. Therefore, it has great meaning in both theory and practice to do comparison studies among these grammars and SMT models to see which of them are capable of better describing parallel translation data. This is a fundamental issue worth exploring in multilingual information processing. However, little effort in previous work has been put in this point. To address this issue, in this paper we define a general platform, namely synchronous tree sequence substitution grammar (STSSG), for the comparison studies. The STSSG can be seen as a generalization of Synchronous TSG (STSG) by replacing elementary tree (a single subtree used in STSG) with contiguous tree sequence as the basic translation unit. As a result, most of previous grammars used in SMT can be interpreted as the reduced versions of the STSSG. Under the STSSG platform, we compare the expressive

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1097–1104
Manchester, August 2008

abilities of various grammars and translation models through linguistically-based synchronous parsing and a real translation platform. By synchronous parsing, we aim to study which grammar can well explain translation data (i.e. translational equivalence alignment) while by the real translation platform, we expect to investigate which model can achieve better translation performance. In addition, we also measure the impact of various factors in this study, including the genera of corpora (newspaper domain via spoken domain), the accuracy of word alignments and syntax parsing (automatically vs. manually).

We report our experimental settings, experimental results and our findings in detail in the rest of the paper, which is organized as follows: Section 2 reviews previous work. Section 3 elaborates the general framework while Section 4 reports the experimental results. Finally, we conclude our work in Section 5.

2 Previous Work

There are only a few of previous work related to the study of translation grammar comparison.

Fox (2002) is the first to look at how well proposed translation models fit actual translation data empirically. She examined the issue of phrasal cohesion between English and French and discovered that while there is less cohesion than one might desire, there is still a large amount of regularity in the constructions where breakdowns occur. This suggests that reordering words by phrasal movement is a reasonable strategy (Fox, 2002). She has also examined the differences in cohesion between Treebank-style parse trees, trees with flattened verb phrases, and dependency structures. Their experimental results indicate that the highest degree of cohesion is present in dependency structures.

Motivated by the same problem raised by Fox (2002), Galley et al. (2004) study what rule can better explain human translation data. They first propose a theory that gives formal semantics to word-level alignments defined over parallel corpora, and then use the theory to introduce a linear algorithm that is used to derive from word-aligned, parallel corpora the minimal set of syntactically motivated transformation rules to explain human translation data. Their basic idea is to create transformation rules that condition on larger fragments of tree structure. Their experimental results suggest that their proposed rules provide a good, realistic indicator of the complexities inherent in translation than SCFG.

Wellington et al. (2006) describes their study of the patterns of translational equivalence exhibited by a variety of bilingual/monolingual bitexts. They empirically measure the lower bounds on alignment failure rates with and without gaps under the constraints of word alignment alone or with one or both side parse trees. Their study finds surprisingly many examples of translational equivalence that could not be analyzed using binary-branching structures without discontinuities. Thus, they claim that the complexity of these patterns in every bitext is higher than suggested in the literature. In addition, they suggest that the low coverage rates without gaps under the constraints of independently generated monolingual parse trees might be the main reason why "syntactic" constraints have not yet increased the accuracy of SMT systems. However, they find that simply allowing a single gap in bilingual phrases or other types of constituent can improve coverage dramatically.

DeNeefe et al. (2007) compares the strengths and weaknesses of a syntax-based MT model with a phrase-based MT model from the viewpoints of translational equivalence extraction methods and coverage. They find that there are surprising differences in phrasal coverage – neither is merely a superset of the other. They also investigate the reason why some phrase pairs are not learned by the syntax-based model. They further propose several solutions and evaluate on the syntax-based extraction techniques in light of phrase pairs captured and translation accuracy. Finally, significant performance improvement is reported using their solutions.

Different from previous work discussed above, this paper mainly focuses on the expressive ability comparison studies among different grammars and models through synchronous parsing and a real SMT platform. Fox (2002), Galley et al (2004) and Wellington et al. (2006) examine TEM only. DeNeefe et al. (2007) only compares the strengths and weaknesses of a syntax-based MT model with a phrase-based MT model.

3 The General Platform: the STSSG

In this section, we first define the STSSG platform in Subsection 3.1, and then explain why it is a general framework that can cover most of previous syntax-based translation grammars and models in Subsection 3.2. In Subsection 3.3 and 3.4, we discuss the STSSG-based SMT and synchronous parsing, which are used to compare different grammars and translation models.

3.1 Definition of the STSSG

The STSSG is an extension of the STSG by using tree sequences (rather than elementary trees) as the basic translation unit. A STSSG is a septet $G = < \Sigma_s, \Sigma_t, N_s, N_t, S_s, S_t, P >$, where:

- Σ_s and Σ_t are source and target terminal alphabets (POSs or lexical words), respectively, and
- N_s and N_t are source and target non-terminal alphabets (linguistic phrase tag, i.e. NP/VP…), respectively, and
- $S_s \in N_s$ and $S_t \in N_t$ are the source and target start symbols (roots of source and target parse trees), and
- P is a production rule set.

A grammar rule r_i in the STSSG is an aligned tree sequence pair, $< \xi_s , \xi_t , \tilde{A} >$, where ξ_s and ξ_t are tree sequences of source side and target sides, respectively, and $\tilde{A}$ is the alignments between leaf nodes of two tree sequences. Here, the key concept of "tree sequence" refers to an ordered subtree sequence covering a consecutive tree fragment in a complete parse tree. The leaf nodes of a subtree in a tree sequence can be either non-terminal symbols or terminal symbols. Fig. 2 shows two STSSG rules extracted from the aligned tree pair shown in Fig. 1, where r_1 is also a STSG rule.

In the STSSG, a translational equivalence is modeled as a tree sequence pair while MT is viewed as a tree sequence substitution process. From the definition of "tree sequence", we can see that a subtree in a tree sequence is a so-called elementary tree used in TSG. This suggests that SCFG and STSG are only a subset of STSSG and SCFG is a subset of STSG. The next subsection discusses how to configure the STSSG to implement the other two simplified grammars. This is the reason why we call the STSSG a general framework for synchronous grammar-based translation modeling.

It is worth noting that, from rule rewriting viewpoint, STSSG can be thought of as a restricted version of synchronous multi-component TAGs (Schuler et al., 2000) although TAG is more powerful than TSG due to the additional operation "adjunctions". The synchronous multi-component TAG can also rewrite several non-terminals in one step of derivation. The difference between them is that the rewriting sites (i.e. the substitution nodes) must be contiguous in STSSG. In addition, STSSG is also related to

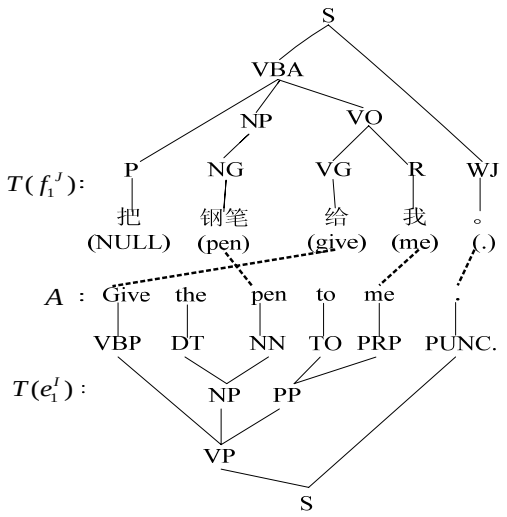

Figure 1. A word-aligned parse tree pairs of a Chinese sentence and its English translation

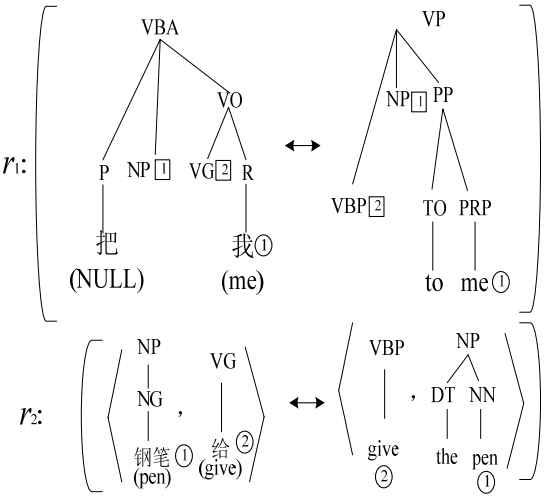

Figure 2. Two examples of translation rules

tree automata (Comon et al., 2007). However, the discussion on the theoretical relation and comparison between them is out of the scope of the paper. In this paper, we focus on the comparison study of SMT grammars using the STSSG platform.

3.2 Rule Extraction and Grammar Configuration

All the STSSG mapping rules are extracted from bi-parsed trees. Our rule extraction algorithm is an extension of that presented at (Chiang, 2005; Liu et al., 2006; Zhang et al., 2007). We modify their tree-to-tree/string rule extraction algorithms to extract tree-sequence-to-tree-sequence rules. Our rules[2] are extracted in two steps:

[2] We classify the rules into two categories: *initial rules*, whose leaf nodes must be terminals, and *ab-*

1) Extracting *initial rules* from bi-parsed trees. This is rather straightforward. We first generate all fully lexicalized source and target tree sequences (whose leaf nodes must be lexical words) using a DP algorithm and then iterate over all generated source and target sequence pairs. If their word alignments are all within the scope of the current tree sequence pair, then the current tree sequence pair is an *initial rule*.

2) Extracting *abstract rules* from the extracted *initial rules*. The idea behind is that we generate an *abstract rule* from a "big" *initial rule* by removing one or more "small" *initial rules* from the "big" one, where the "small" ones must be a sub-graph of the "big" one. Please refer to (Chiang, 2005; Liu et al., 2006; Zhang et al., 2007) for the implementation details.

As indicated before (Chiang, 2005; Zhang et al., 2007), the above scheme generates a very large number of rules, which not only makes the system too complicated but also introduces too many undesirable ambiguities. To control the overall model complexity, we introduce the following parameters:

1) The maximal numbers of trees in the source and target tree sequences: α_s and α_t.

2) The maximal tree heights in the source and target tree sequences: β_s and β_t.

3) The maximal numbers of non-terminal leaf nodes in the source and target tree sequences: γ_s and γ_t.

Now let us see how to implement other models in relation to STSSG based the STSSG through configuring the above parameters.

1) STSG-based tree-to-tree model (Zhang et al., 2007; Bod, 2007) when $\alpha_s = \alpha_t = 1$.

2) SCFG-based tree-to-tree model when $\alpha_s = \alpha_t = 1$ and $\beta_s = \beta_t = 2$.

3) Phrase-based translation model only (no reordering model) when $\gamma_s = \gamma_t = 0$ and $\beta_s = \beta_t = 1$.

4) TSG-CFG-based tree-to-string model (Liu et al., 2006) when $\alpha_s = \alpha_t = 1$, $\beta_t = 2$ and ignore phrase tags in target side.

5) CFG-TSG-based string-to-tree model (Galley et al., 2006) when $\alpha_s = \alpha_t = 1$and $\beta_s = 2$.

6) TSSG-CFG-based tree-sequence-to-string model (Liu et al., 2007) when $\beta_t = 2$ and ignore phrase tags in target side.

From the above definitions, we can see that all of previous related models/grammars can be can be interpreted as the reduced versions of the STSSG. This is the reason why we use the STSSG as a general platform for our model and grammar comparison studies.

3.3 Model Training and Decoder for SMT

We use the tree sequence mapping rules to model the translation process. Given the source parse tree $T(f_1^J)$, there are multiple derivations[3] that could lead to the same target tree $T(e_1^I)$, the mapping probability $Pr(T(e_1^I)|T(f_1^J))$ is obtained by summing over the probabilities of all derivations. The probability of each derivation θ is given by the product of the probabilities of all the rules $p(r_i)$ used in the derivation (here we assume that a rule is applied *independently* in a derivation).

$$\begin{aligned} Pr(e_1^I \mid f_1^J) &= Pr(T(e_1^I)|T(f_1^J)) \\ &= \sum_{\theta} \prod_{r_i \in \theta} p(r_i) \end{aligned} \quad (1)$$

The model is implemented under log-linear framework. We use seven basic features that are analogous to the commonly used features in phrase-based systems (Koehn, 2004): 1) bidirectional rule mapping probabilities; 2) bidirectional lexical translation probabilities; 3) the target language model; 4) the number of rules used and 5) the number of target words. Besides, we define two new features: 1) the number of lexical words in a rule to control the model's preference for lexicalized rules over un-lexicalized rules and 2) the average tree height in a rule to balance the usage of hierarchical rules and more flat rules. The overall training process is similar to the process in the phrase-based system (koehn et al., 2007): word alignment, rule extraction, feature extraction and probability calculation and feature weight tuning.

Given $T(f_1^J)$, the decoder is to find the best derivation θ that generates $<T(f_1^J), T(e_1^I)>$.

$$\begin{aligned} \hat{e} &= \arg\max_{e_1^I} Pr(T(e_1^I)|T(f_1^J)) \\ &\approx \arg\max_{e_1^I,\theta} \prod_{r_i \in \theta} p(r_i) \end{aligned} \quad (2)$$

By default, same as other SMT decoder, here we use Viterbi derivation in Eq (2) instead of the

stract rule that having at least one non-terminal leaf node.

[3] A derivation is a sequence of tree sequence rules that maps a source parse tree to its target one.

summing probabilities in Eq (3). This is to make the decoder speed not too slow. The decoder is a standard span-based chart parser together with a function for mapping the source derivations to the target ones. To speed up the decoder, we utilize several thresholds to limit the search beams for each span, such as the number of rules used and the number of hypotheses generated.

3.4 Synchronous Parsing

A synchronous parser is an algorithm that can infer the syntactic structure of each component text in a multitext and simultaneously infer the correspondence relation between these structures. When a parser's input can have fewer dimensions than the parser's grammar, we call it a translator. When a parser's grammar can have fewer dimensions than the parser's input, we call it a synchronizer (Melamed, 2004). Therefore, synchronous parsing and MT are closed to each other. In this paper, we use synchronous parsing to compare the ability of different grammars in translational equivalence modeling.

Given a bilingual sentence pair f_1^J and e_1^I, the synchronous parser is to find a derivation θ that generates $< T(f_1^J), T(e_1^I) >$. Our synchronous parser is similar to the synchronous CKY parser presented at (Melamed, 2004). The difference is that we implement it based on our STSSG decoder. Therefore, in nature the parser is a standard synchronous chart parser but constrained by the rules of the STSSG grammar. In our implementation, we simply use our decoder to simulate the bilingual parser: 1) for each sentence pair, we extract one model; 2) we use the model and the decoder to translate the source sentence of the given sentence pair; 3) if the target sentence is successfully generated by the decoder, then we say the symphonious parsing is successful. Please note that the synchronous parsing is considered as successful once the last words in the source and target sentences are covered by the decoder even if there is no a complete target parse tree generated (it may be a tree sequence). This is because our study only concerns whether all translational equivalences are linked together by the synchronous parser correctly.

4 Experiments

4.1 Experimental Settings

Synchronous parsing settings: Our experiments of synchronous parsing are carried on three Chinese-to-English bilingual corpora: the FBIS corpus, the IWSLT 2007 training set and the HIT Corpus. The FBIS data is a collection of translated newswire documents published by major news agencies from three representative locations: Beijing, Taipei and Hongkong. The IWSLT data is a multilingual speech corpus on travel domain while the HIT corpus consists of example sentences of a Chinese-English dictionary. The first two corpora are sentence-aligned while the HIT corpus is a manually bi-parsed corpus with manually annotated word alignments. We use the three corpora to study whether the models' expressive abilities are domain dependent and how the performance of word alignment and parsing affect the ability of translation models. We selected 2000 sentence pairs from each individual corpus for the comparison study of translational equivalence modeling. Table 1 gives descriptive statistics of the tree data set.

	Chinese	English
FBIS	48,331	59,788
IWSLT	17,667	18,427
HIT	18,215	20,266

Table 1. # of words of experimental data for synchronous parsing (there are 2k sentence pairs in each individual corpus)

In the synchronous parsing experiments, we compared three synchronous grammars: SCFG, STSG and STSSG using the STSSG platform. We use the same settings except the following parameters (please refer to Subsection 3.2 for their definitions): $\alpha_s = \alpha_t = 1$, $\beta_s = \beta_t = 2$ for SCFG ; $\alpha_s = \alpha_t = 1$ and $\beta_s = \beta_t = 6$ for STSG; $\alpha_s = \alpha_t = 4$ and $\beta_s = \beta_t = 6$ for STSSG. We iterate over each sentence pair in the three corpora with the following process:

1) to used Stanford parser (Klein and Manning, 2003) to parse bilingual sentences separately, this means that our study is based on the Penn Treebank style grammar.

2) to extract SCFG, STSG and STSSG rules form each sentence pair, respectively;

3) to do synchronous parsing using the exacted rules.

Finally, we can calculate the successful rate of the synchronous parsing on each corpus.

SMT evaluation settings: For the SMT experiments, we trained the translation model on the FBIS corpus (7.2M (Chinese)+9.2M(English) words) and trained a 4-gram language model on

the Xinhua portion of the English Gigaword corpus (181M words) using the SRILM Toolkits (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1998). We used these sentences with less than 50 characters from the NIST MT-2002 test set as our development set and the NIST MT-2005 test set as our test set. We used the Stanford parser (Klein and Manning, 2003) to parse bilingual sentences on the training set and Chinese sentences on the development and test sets. The evaluation metric is case-sensitive BLEU-4 (Papineni et al., 2002). We used GIZA++ and the heuristics "grow-diag-final" to generate m-to-n word alignments. For the MER training, we modified Koehn's MER trainer (Koehn, 2004) for our STSSG-based system. For significance test, we used Zhang et al's implementation (Zhang et al, 2004). We compared four SMT systems: Moses (Koehn et al., 2007), SCFG-based, STSG-based and STSSG-based tree-to-tree translation models. For Moses, we used its default settings. For the others, we implemented them on the STSSG platform by adopting the same settings as used in the synchronous parsing. We optimized the decoding parameters on the development sets empirically.

4.2 Experimental Results

	SCFG	STSG	STSSG
FBIS	7 (0.35%)	143 (7.15%)	388 (19.4%)
IWSLT	171 (8.6%)	1179 (58.9%)	1708 (85.4%)
HIT	65 (3.23%)	1133 (56.6%)	1532 (76.6%)

Table 2. Successful rates (numbers inside bracket) of synchronous parsing over 2,000 sentence pairs, where the integers outside bracket are the numbers of successfully-parsed sentence pairs

Table 2 reports the experimental results of synchronous parsing. It shows that:

1) As an extension of STSG/SCFG, STSSG outperforms STSG and SCFG consistently in the three data sets. The significant difference suggests that the STSSG is much more effective in modeling translational equivalences and structure divergences. The reason is simply because the STSSG uses tree sequences as the basic translation unit so that it can model non-syntactic phrase equivalence with structure information and handle structure reordering in a large span.

2) STSG shows much better performance than SCFG. It is mainly due to that STSG allow multiple level tree nodes operation and reordering in a larger span than SCFG. It reconfirms that only allowing sibling nodes reordering as done in SCFG may be inadequate for translational equivalence modeling (Galley et al., 2004)[4].

3) All the three models on the FBIS corpus show much lower performance than that on the other two corpora. The main reason, as shown in Table 1, is that the sentences in the FBIS corpus are much longer than that in the other corpus, so their syntactic structures are significantly more complicated than the other two. In addition, although tree sequences are utilized, STSSG show much lower performance in the FBIS corpus. This implies that the complexity of structure divergence between two languages is higher than suggested in literature (Fox, 2002; Galley et al., 2004). Therefore, structure divergence is still a big challenge to translational equivalence modeling when using syntactic structure mapping.

4) The HIT corpus does not show better performance than the IWSLT corpus although the HIT corpus is manually annotated with parse trees and word alignments. In order to study whether high performance word alignment and parsing results can help synchronous parsing, we do several cross validations and report the experimental results in Table 3.

	Gold Word Alignment	Automatic Word Alignment
Gold Parse	3.2/56.6/76.6	2.9/57.7/80.9
Automatic Parse	3.2/55.6/76.0	2.9/54.2/78.8

Table 3. Successful rates (SCFG/STSG/STSSG)(%) with regards to different word alignments and parse trees on the HIT corpus

Table 3 compares the performance of synchronous parsing on the HIT corpus when using gold and automatic parser and word alignment. It is surprised that gold word alignments and parse trees do not help and even decrease the performance slightly. Our analysis further finds that

[4] This claim is mainly hold for linguistically-informed SCFG since formal SCFG and BTG already showed much better performance in the formally syntax-based translation framework (Chiang, 2005). This is because the formal syntax is learned from phrase translational equivalences directly without relying on any linguistic theory (Chiang, 2005). Thus, it may not suffer from the issues of non-isomorphic structure alignment and non-syntactic phrase usage heavily (Wellington et al., 2006).

more than 90% sentence pairs out of all the sentence pairs that can be successfully bi-parsed are in common in the four experiments. This suggests that the STSSG/STSG (SCFG achieves too much lower performance) and our rule extraction algorithm are robust in dealing with the errors introduced by the word alignment and parsing programs. If a parser, for example, makes a systematic error, we expect to learn a rule that can nevertheless be systematically used to model correct translational equivalence. Our error analysis on the three corpora shows that most of the failures of synchronous parsing are due to the structure divergence (i.e. the nature of non-isomorphic structure mapping) and the long distance dependence in the syntactic structures.

	SCFG	Moses	STSG	STSSG
BLEU(%)	22.72	23.86	24.71	26.07

Table 3. Performance comparison of different grammars on FBIS corpus

Table 3 compares different grammars in terms of translation performance. It shows that:

1) The same as synchronous parsing, the STSSG-based model statistically significantly outperforms ($p < 0.01$) previous phrase-based and linguistically syntax-based methods. This empirically verifies the effect of the tree-sequence-based grammar for statistical machine translation.

2) Both STSSG and STSG outperform Moses significantly and STSSG clearly outperforms STSG, which suggest that:

- The linguistically motivated structure features are still useful for SMT, which can be captured by the two syntax-based grammars through tree node operations.
- STSSG is much more effective in utilizing linguistic structures than STSG since it uses tree sequence as the basic translation unit. This enables STSSG not only to handle structure reorderings by tree node operations in a larger span, but also to capture non-syntactic phrases with syntactic information, and hence giving the grammar more expressive power.

3) The linguistic-based SCFG shows much lower performance. This is largely because SCFG only allows sibling nodes reordering and fails to utilize both non-syntactic phrases and those syntactic phrases that cannot be covered by a single CFG rule. It thereby suggests that SCFG is less effective in modelling parse tree structure transfer.

The above two experimental results show that STSSG achieves significant improvements over the other two grammars in terms of synchronous parsing's successful rate and translation Bleu score.

5 Conclusions

Grammar is the fundamental infrastructure in translational equivalence modeling and statistical machine translation since grammar formalizes what kind of rule to be learned from a parallel text. In this paper, we first present a general platform STSSG and demonstrate that a number of synchronous grammars and SMT models can be easily implemented based on the platform. We then compare the expressive abilities of different grammars on the platform using synchronous parsing and statistical machine translation. Our experimental results show that STSSG can better explain the data in parallel corpora than the other two synchronous grammars. We further finds that, although syntactic structure features are helpful in modeling translational equivalence, the complexity of structure divergence is much higher than suggested in literature, which imposes a big challenge to syntactic transformation-based SMT. This may explain why traditional syntactic constraints in SMT do not yield much performance improvement over robust phrase-substitution models.

The fundamental assumption underlying much recent work on syntax-based modeling, which is considered to be one of next technology breakthroughs in SMT, is that translational equivalence can be well modeled by structural transformation. However, as discussed in prior arts (Galley et al., 2004) and this paper, linguistically-informed SCFG is an inadequate model for parallel corpora due to its nature that only allowing child-node reorderings. Although STSG shows much better performance than SCFG, its two major limitations are that it only allows structure distortion operated on a single sub-tree and cannot model non-syntactic phrases. STSSG extends STSG by using tree sequence as the basic translation unit. This gives the grammar much more expressive power.

There are many open issues in the syntactic transformation-based SMT due to the divergence nature between bilingual structure mappings. We find that structural divergences are more serious than suggested in the literature (Fox, 2002; Gallery et al., 2004) or what we expected when sentences are longer. We will continue to investigate

whether and how parallel corpora can be well modeled by syntactic structure mappings.

References

Rens Bod. 2007. *Unsupervised Syntax-Based Machine Translation: The Contribution of Discontinuous Phrases.* MT-Summmit-07. 51-56.

Peter F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. *The mathematics of machine translation: Parameter estimation*. Computational Linguistics, 19(2):263–311.

S. F. Chen and J. Goodman. 1998. *An empirical study of smoothing techniques for language modeling*. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

David Chiang. 2005. *A hierarchical phrase-based model for SMT*. ACL-05. 263-270.

H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. 2007. Tree automata techniques and applications. Available at: http://tata.gforge.inria.fr/.

Brooke Cowan, Ivona Kucerova and Michael Collins. 2006. *A discriminative model for tree-to-tree translation*. EMNLP-06. 232-241.

S. DeNeefe, K. Knight, W. Wang and D. Marcu. 2007. *What Can Syntax-based MT Learn from Phrase-based MT?* EMNLP-CoNLL-07. 755-763

Yuan Ding and Martha Palmer. 2005. *Machine translation using probabilistic synchronous dependency insertion grammars*. ACL-05. 541-548.

Bonnie J. Dorr (1994). *Machine Translation Divergences: A formal description and proposed solution*. Computational Linguistics, 20(4): 597-633

Jason Eisner. 2003. *Learning non-isomorphic tree mappings for MT*. ACL-03 (companion volume).

Heidi J. Fox. 2002. *Phrasal Cohesion and Statistical Machine Translation*. EMNLP-2002. 304-311

Michel Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang and I. Thayer. 2006. *Scalable Inference and Training of Context-Rich Syntactic Translation Models.* COLING-ACL-06. 961-968

M. Galley, M. Hopkins, K. Knight and D. Marcu. 2004. *What's in a translation rule?* HLT-NAACL.

Liang Huang, Kevin Knight and Aravind Joshi. 2006. *Statistical Syntax-Directed Translation with Extended Domain of Locality*. AMTA-06 (poster).

Mary Hearne and Andy Way. 2003. *Seeing the wood for the trees: data-oriented translation*. MT Summit IX, 165-172.

Dan Klein and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. ACL-03. 423-430.

Philipp Koehn, F. J. Och and D. Marcu. 2003. *Statistical phrase-based translation.* HLT-NAACL-03. 127-133.

Philipp Koehn. 2004. *Pharaoh: a beam search decoder for phrase-based statistical machine translation models*. AMTA-04, 115-124.

Philipp Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst. 2007. *Moses: Open Source Toolkit for Statistical Machine Translation*. ACL-07 (poster) 77-180.

Yang Liu, Qun Liu and Shouxun Lin. 2006. *Tree-to-String Alignment Template for Statistical Machine Translation*. COLING-ACL-06. 609-616.

Yang Liu, Yun Huang, Qun Liu and Shouxun Lin. 2007. *Forest-to-String Statistical Translation Rules*. ACL-07. 704-711.

Daniel Marcu, W. Wang, A. Echihabi and K. Knight. 2006. *SPMT: Statistical Machine Translation with Syntactified Target Language Phrases*. EMNLP-06. 44-52.

I. Dan Melamed. 2004. *Statistical machine translation by parsing*. ACL-04. 653-660.

K. Papineni, Salim Roukos, ToddWard and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. ACL-02. 311-318.

Arjen Poutsma. 2000. *Data-oriented translation. COLING-2000*. 635-641

Chris Quirk, Arul Menezes and Colin Cherry. 2005. *Dependency treelet translation: Syntactically informed phrasal SMT*. ACL-05. 271-279.

William Schuler, David Chiang and Mark Dras. 2000. *Multi-Component TAG and Notions of Formal Power*. ACL-2000. 448-455

Andreas Stolcke. 2002. *SRILM - an extensible language modeling toolkit*. ICSLP-02. 901-904.

Benjamin Wellington, Sonjia Waxmonsky and I. Dan Melamed. 2006. *Empirical Lower Bounds on the Complexity of Translational Equivalence*. COLING-ACL-06. 977-984.

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

K. Yamada and Kevin Knight. 2001. *A syntax-based statistical translation model*. ACL-01. 523-530.

M. Zhang, H. Jiang, A. Aw, J. Sun, S. Li and C. Tan. 2007. *A Tree-to-Tree Alignment-based Model for SMT*. MT-Summit-07. 535-542.

Y. Zhang, S. Vogel and A. Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? LREC-04.