Deep Syntactic Structures for String-to-Tree Translation

Xianchao Wu[†] **Jun'ichi Tsujii**[†‡]

[†]Computer Science, Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[‡]School of Computer Science, University of Manchester
National Centre for Text Mining (NaCTeM)
Manchester Interdisciplinary Biocentre, 131 Princess Street, Manchester M1 7DN, UK

{wxc, tsujii}@is.s.u-tokyo.ac.jp

1 Introduction

1.1 String-to-tree translation

A state-of-the-art syntax-based Statistical Machine Translation (SMT) model, string-to-tree translation model (Galley et al., 2004; Galley et al., 2006; Chiang et al., 2009), is to construct a number of parse trees of the target language by 'parsing' a source language sentence making use of a bilingual translation grammar. Given a set of parallel sentences for training, optimal word alignments for every sentence pairs are first derived by using GIZA++ (Och and Ney, 2003). Then, a syntactic parser is used to parse the target sentences into trees. A source sentence, a target parse tree, and the alignment between the source and target words form an 'aligned tree-string pair'.

In order to split a tree into tree fragments yet obeying the constraints from the word alignment, algorithms proposed by Galley et al. (2004; 2006) are the de facto standard for extracting minimal and composed tree-string translation rules from the aligned tree-string pairs. n-gram language model (LM) integrated CKY algorithm (Huang and Chiang, 2005; Chiang, 2007) is popularly used for decoding. Tree-string translation rules are binarized (Zhang et al., 2006) into Chomsky normal forms before been used by the CKY algorithm.

Figure 1 illustrates the training and testing process of Japanese string to English tree translation. For simplicity, similar example is used for both translation rule extracting and decoding. During training, suppose we are given an aligned tree-string pair, GHKM algorithm (Galley et al., 2004) is first applied for minimal rule extraction. During testing, given a Japanese sentence, we try to build a number of English parse trees using the (binarized) translation rules. The translation output can be easily collected by accessing the leaves in a parse tree through a left-to-right traversal.

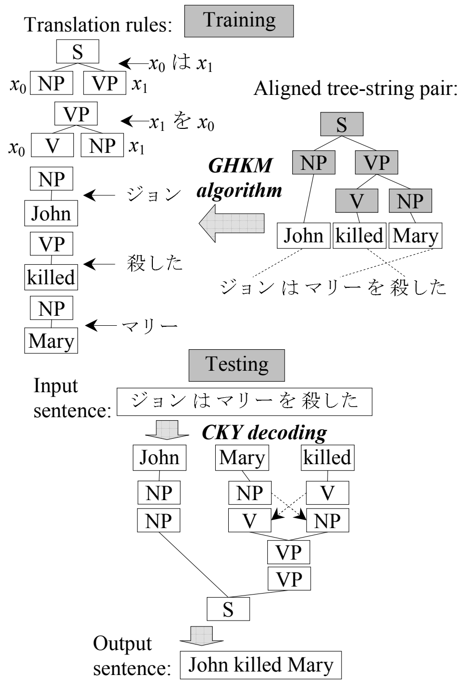

Figure 1: Illustration of the training and testing process for string-to-tree translation.

1.2 Deep syntactic structures

In contrast to commit to a Probabilistic Context-Free Grammar (PCFG) parser which only generates shallow trees of English (Galley et al., 2004; Galley et al., 2006; Chiang et al., 2009), we propose the use of *deep* parse trees and *semantic dependencies* described respectively by Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994) and

feature	*description*
CAT	phrasal category
XCAT	fine-grained phrasal category
SCHEMA	name of the schema applied in the node
HEAD	*pointer* to the head daughter
SEM_HEAD	*pointer* to the semantic head daughter
CAT	syntactic category
POS	Penn Treebank-style part-of-speech tag
BASE	base form
TENSE	tense of a verb
ASPECT	aspect of a verb
VOICE	voice of a verb
AUX	auxiliary verb or not
LEXENTRY	assigned lexical entry
PRED	type of a predicate
ARG$\langle x \rangle$	*pointer* to semantic arguments, $x = 1..4$

Table 1: Syntactic/semantic features extracted from Enju's HPSG signs.

Predicate-Argument Structures (PASs).

We illustrate two major characteristics that an HPSG tree yielded by Enju[1], a state-of-the-art HPSG parser for English, differs from a traditional PCFG tree. First, a node in an HPSG tree is represented by a *typed feature structure* (TFS) with richer information (Table 1) than a PCFG node that is commonly represented by only POS/phrasal tags. Second, PASs, which describe the semantic relations among a predicate (can be a verb, adjective, preposition, etc.) and its arguments, are used for guiding local/global reordering during translation.

For example, in Figure 2, we show the HPSG pare tree of the English sentence *John killed Mary*. Each node in the tree is expressed by a set of instantiated features as listed in Table 1. Consequently, each tree-string rule listed in the left-bottom corner of Figure 2 include an HPSG tree fragment. For simplicity, we only use the node identifiers such as c_0, t_0 to represent the complete TFSs in the tree. Also, transitive verb *killed* has a PRED to be 'verb_arg12', the subject argument ARG1 to be c_1, and the direct object argument ARG2 to be c_5. In order to localize this semantic dependency into one tree fragment, we define *minimum covering tree* to cover a predicate word and its arguments. The tree-string rule listed in the right-bottom corner of Figure 2 reflects that two arguments are necessary for the transitive verb *killed*. The reordering among *killed* and its two arguments are reflected as well by the alignments in broken lines.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/index.html

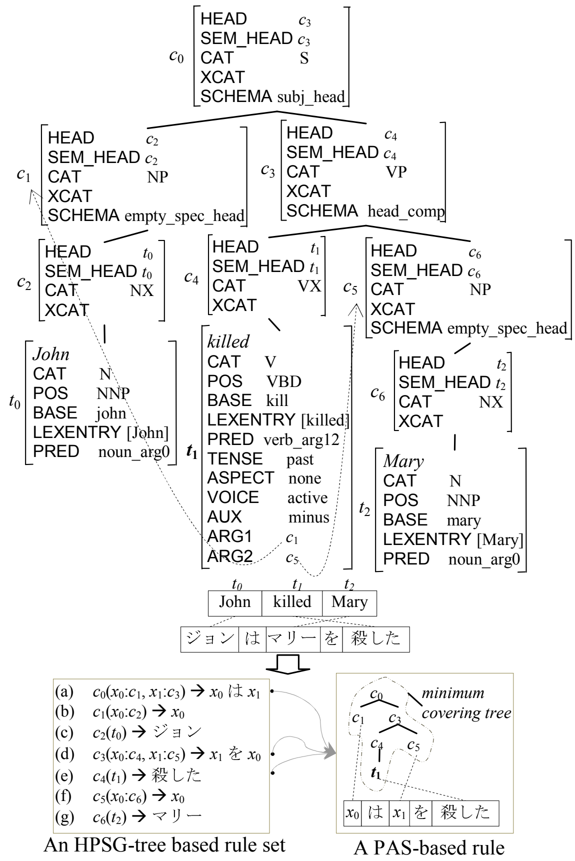

Figure 2: Illustration of the deep syntactic structures for rule extraction.

2 Binarization

For efficient decoding with integrated n-gram LMs, we follow (Zhang et al., 2006) to synchronously binarize all translation rules (extracted from the HPSG-tree structures and PASs) into Chomsky Normal Forms that contain at most two variables and can be incrementally scored by LM. In order to make use of the binarized rules in the CKY decoding, we add two kinds of glues rules:

$$S \rightarrow X_m{}^{(1)}, X_m{}^{(1)}; \quad (1)$$

$$S \rightarrow S^{(1)} X_m{}^{(2)}, S^{(1)} X_m{}^{(2)}. \quad (2)$$

Here X_m ranges over the nonterminals that appear in the binarized rule set. These glue rules can be seen as an extension from X to $\{X_m\}$of the two glue rules described in (Chiang, 2007).

3 Translation model and decoding

Our string-to-tree model utilizes a (Hiero-style) phrase-translation table (PTT) generated by using Moses (Koehn et al., 2007), a (binarized) HPSG tree-based rule set (TRS) extracted by using the algorithms described in (Galley et al., 2006), and a (binarized) PAS-based rule set (PRS) extracted by using the Algorithm 1 described in (Wu et al., 2009). We use Z-mert[2] (Zaidan, 2009) to tune the weights of the features from PTT, TRS, and PRS on the development set.

The decoder searches for the optimal derivation d^* that transforms a source (e.g., Japanese) sentence F into a parse forest of English among the set of all possible derivations D:

$$d^* = \arg\max_{d \in D}\{\lambda_1 \log p_{LM}(\tau(d)) + \lambda_2|\tau(d)| \quad (3)$$

$$+ \lambda_3 g(d) + \log s(d|F)\}. \quad (4)$$

Here, the first item is the LM probability, the second item is the translation length penalty where $\tau(d)$ is the target string for derivation d, the third item is the number of glue rules used in d, and the forth item is the translation score, which is further decomposed into the product of rule feature values:

$$s(d|F) = \prod_{r \in d} f(r_1) f(r_2) f(r_3), \quad (5)$$

where $r_1 \in$ PTT, $r_2 \in$ TRS, and $r_3 \in$ PRS. This equation reflects that the translation rules come from three sets. Each $f(r)$ is in turn a product of five feature functions:

$$f(r) = p(s|t)^{\lambda_4} \cdot p(t|s)^{\lambda_5} \cdot l(s|t)^{\lambda_6} \cdot l(t|s)^{\lambda_7} \cdot e^{\lambda_8}. \quad (6)$$

Here, s/t represent the source/target phrases of a rule in PTT, TRS, or PRS; $p(\cdot|\cdot)$ and $l(\cdot|\cdot)$ are the translation and lexical probabilities of rules from PTT, TRS, and PRS. Note that the derivation length penalty is controlled by λ_8.

We use a CKY-style algorithm with beam-pruning and cube-pruning (Chiang, 2007) to decode Chinese sentences. For each source language sentence F, the output of the chart-parsing algorithm is expressed as a *hyper-graph* representing a set of derivations. Given such a hyper-graph, we use the Algorithm 3 described in (Huang and Chiang, 2005) to extract its k (= 200) best derivations for MERT.

[2]http://www.cs.jhu.edu/ ozaidan/zmert/

system	*BLEU(%)*	*node type*
Joshua	14.00	-
PTT	12.98	-
PTT+PRS	14.65 *	TFS
PTT+C_3^S	15.23 **	POS/phrasal
PTT+C_3	15.83 **	TFS
PTT+C_3+PRS	15.85 **	TFS

Table 2: The BLEU scores achieved by Joshua and our system variants. * or ** = significantly better than Joshua ($p < 0.05$ or 0.01, respectively).

4 Experiments

The JST Japanese-English paper abstract corpus[3], which consists of one million parallel sentences, was used for training and testing. Using Enju2.3.1, we successfully parsed 987,401 English sentences of the 994K sentences in the training set, with a success rate of 99.3%. Both the development and the test set contain 2K parallel sentences.

For HPSG-tree based rule extraction, we follow (Galley et al., 2006) to construct derivation-forests for each aligned tree-string pairs in order to include rich syntactic context and cover the feasible attachments of unaligned Japanese words. There are still 6.3% unaligned Japanese words appearing in 83.7% of the training sentences after using GIZA++ and *grow-diag-final-and* (Koehn et al., 2007) balancing strategy. SRILM toolkit (Stolcke, 2002) was employed to train a 5-gram LM on the 994K English sentences with modified Kneser-Ney smoothing.

The baseline system for comparison is Joshua (Li et al., 2009), a freely available decoder which implemented the hierarchical phrase-based translation model (Chiang, 2005). We evaluated the translation quality using the BLEU metric (Papineni et al., 2002). We used four dual core Xeon machines (4×3.0GHz×2CPU, 4×64GB memory) to run all the experiments.

The translation accuracies of Joshua and our system variants are shown in Table 2. C_3 represents the set of composed rules in which the number of internal nodes in tree fragments (of tree-string rules) is no more than 3. C_3^S is similar with C_3 except the TFS of each node is replaced by POS/phrasal tags.

No reordering was performed when using PTT for decoding, which explains that only use PTT in our

[3]http://www.jst.go.jp

system performed worse than Joshua. We gained a significant improvement ($p < 0.01$) on BLEU score by appending PRS to PTT. This reflects that PAS-based rules are compact and helpful for reordering. PTT+PRS is significantly better ($p < 0.05$) than Joshua, thanks to the TFS and semantic information included in the PAS-based rules.

By replacing simple POS/phrasal tags with TFSs, we gained 0.6 (%) BLEU points from PTT+C_3^S to PTT+C_3. This tells that HPSG trees do perform a fine-grained description of the syntactic property.

Finally, by appending PRS to PTT+C_3, the BLEU score changed slightly. We argue this is because C_3 covers most rules in PRS. For example, the PAS-based rule listed in the right-bottom corner of Figure 2 is also a composed rule ($\in C_3$) by connecting the three minimal rules (a), (d), and (e) listed in the left-bottom corner of Figure 2 (gray arrows).

5 Conclusion

We have introduced deep syntactic structures which significantly improved the performance of string-to-tree translation. We used an HPSG parser to obtain the deep syntactic structure of a target sentence, which includes fine-grained description of the syntactic property and also a semantic representation of the sentence. We described the log-linear translation model and n-gram LM integrated CKY-decoding of our string-to-tree system. Experiments on large-scale Japanese-to-English translation testified the significant effectiveness of our proposal. The system will be released as an open source toolkit in the near future.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan) and Japanese/Chinese Machine Translation Project in Special Coordination Funds for Promoting Science and Technology (MEXT, Japan), and Microsoft Research Asia Machine Translation Theme. The first author thanks Takuya Matsuzaki, Naoaki Okazaki, and Yusuke Miyao for their invaluable suggestions and help.

References

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of HLT-NAACL*, pages 218–226, June.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270, Ann Arbor, MI.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Lingustics*, 33(2):201–228.

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

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL*, pages 961–968, Sydney.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of IWPT*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL Demo*, pages 177–180.

Zhifei Li, Chris Callison-Burch, Chris Dyery, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. 2009. Demonstration of joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 25–28, Suntec, Singapore, August.

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

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

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of ICSLP*.

Xianchao Wu, Takuya Matsuzaki, Naoaki Okazaki, Yusuke Miyao, and Jun'ichi Tsujii. 2009. The UOT System: Improve String-to-Tree Translation Using Head-Driven Phrasal Structure Grammar and Predicate-Argument Structures. In *Proceedings of IWSLT*, pages 99–106, Tokyo, Japan.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of HLT-NAACL*, pages 256–263, New York City, USA, June.