Syntax-Augmented Machine Translation using Syntax-Label Clustering

Hideya Mino, Taro Watanabe and **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, JAPAN
{hideya.mino, taro.watanabe, eiichiro.sumita}@nict.go.jp

Abstract

Recently, syntactic information has helped significantly to improve statistical machine translation. However, the use of syntactic information may have a negative impact on the speed of translation because of the large number of rules, especially when syntax labels are projected from a parser in syntax-augmented machine translation. In this paper, we propose a syntax-label clustering method that uses an exchange algorithm in which syntax labels are clustered together to reduce the number of rules. The proposed method achieves clustering by directly maximizing the likelihood of synchronous rules, whereas previous work considered only the similarity of probabilistic distributions of labels. We tested the proposed method on Japanese-English and Chinese-English translation tasks and found order-of-magnitude higher clustering speeds for reducing labels and gains in translation quality compared with previous clustering method.

1 Introduction

In recent years, statistical machine translation (SMT) models that use syntactic information have received significant research attention. These models use syntactic information on the source side (Liu et al., 2006; Mylonakis and Sima'an, 2011), the target side (Galley et al., 2006; Huang and Knight, 2006) or both sides (Chiang, 2010; Hanneman and Lavie, 2013) produce syntactically correct translations. Zollmann and Venugopal (2006) proposed syntax-augmented MT (SAMT), which is a MT system that uses syntax labels of a parser. The SAMT grammar directly encodes syntactic information into the synchronous context-free grammar (SCFG) of Hiero (Chiang, 2007), which relies on two nonterminal labels. One problem in adding syntax labels to Hiero-style rules is that only partial phrases are assigned labels. It is common practice to extend labels by using the idea of combinatory categorial grammar (CCG) (Steedman, 2000) on the problem. Although this extended syntactical information may improve the coverage of rules and syntactic correctness in translation, the increased grammar size causes serious speed and data-sparseness problems. To address these problems, Hanneman and Lavie (2013) coarsen syntactic labels using the similarity of the probabilistic distributions of labels in synchronous rules and showed that performance improved.

In the present work, we follow the idea of label-set coarsening and propose a new method to group syntax labels. First, as an optimization criterion, we use the logarithm of the likelihood of synchronous rules instead of the similarity of probabilistic distributions of syntax labels. Second, we use exchange clustering (Uszkoreit and Brants, 2008), which is faster than the agglomerative-clustering algorithm used in the previous work. We tested our proposed method on Japanese-English and Chinese-English translation tasks and observed gains comparable to those of previous work with similar reductions in grammar size.

2 Syntax-Augmented Machine Translation

SAMT is an instance of SCFG $\mathcal{G}$, which can be formally defined as

$$\mathcal{G} = (\mathcal{N}, S, \mathcal{T}_\sigma, \mathcal{T}_\tau, \mathcal{R})$$

where $\mathcal{N}$ is a set of nonterminals, $S \in \mathcal{N}$ is a start label, $\mathcal{T}_\sigma$ and $\mathcal{T}_\tau$ are the source- and target-side terminals, and $\mathcal{R}$ is a set of synchronous rules. Each synchronous rule in $\mathcal{R}$ takes the form

$$X \rightarrow \langle \alpha, \beta, \sim \rangle$$

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 165–171,
October 25-29, 2014, Doha, Qatar.

where $X \in \mathcal{N}$ is a nonterminal, $\alpha \in (\mathcal{N} \cup \mathcal{T}_\sigma)*$ is a sequence of nonterminals or source-side terminals, and $\beta \in (\mathcal{N} \cup \mathcal{T}_\tau)*$ is a sequence of nonterminals or target-side terminals. The number $\#NT(\alpha)$ of nonterminals in α is equal to the number $\#NT(\beta)$ of nonterminals in β, and $\sim$: $\{1, ..., \#NT(\alpha)\} \rightarrow \{1, ..., \#NT(\beta)\}$ is a one-to-one mapping from nonterminals in α to nonterminals in β. For each synchronous rule, a nonnegative real-value weight $w(X \rightarrow \langle \alpha, \beta, \sim \rangle)$ is assigned and the sum of the weights of all rules sharing the same left-hand side in a grammar is unity.

Hierarchical phrase-based SMT (Hiero) (Chiang, 2007) translates by using synchronous rules that only have two nonterminal labels X and S but have no linguistic information. SAMT augments the Hiero-style rules with syntax labels from a parser and extends these labels based on CCG. Although the use of extended syntax labels may increase the coverage of rules and improve the potential for syntactically correct translations, the growth of the nonterminal symbols significantly affects the speed of decoding and causes a serious data-sparseness problem.

To address these problems, Hanneman and Lavie (2013) proposed a label-collapsing algorithm, in which syntax labels are clustered by using the similarity of the probabilistic distributions of clustered labels in synchronous rules. First, Hanneman and Lavie defined the label-alignment distribution as

$$P(s|t) = \frac{\#(s,t)}{\#(t)} \tag{1}$$

where $\mathcal{N}_\sigma$ and $\mathcal{N}_\tau$ are the source- and target-side nonterminals in synchronous rules, $s \in \mathcal{N}_\sigma$ and $t \in \mathcal{N}_\tau$ are syntax labels from the source and target sides, $\#(s,t)$ denotes the number of left-hand-side label pairs, and $\#(t)$ denotes the number of target-side labels. Second, for each target-side label pair (t_i, t_j), we calculate the total distance d of the absolute differences in the likelihood of labels that are aligned to a source-side label s:

$$d(t_i, t_j) = \sum_{s \in \mathcal{N}_\sigma} |P(s|t_i) - P(s|t_j)| \tag{2}$$

Next, the closest syntax-label pair of $\hat{t}$ and $\hat{t}'$ is combined into a new single label. The agglomerative clustering is applied iteratively until the number of the syntax labels reaches a given value.

The clustering of Hanneman and Lavie proved successful in decreasing the grammar size and providing a statistically significant improvement in translation quality. However, their method relies on an agglomerative clustering with a worst-case time complexity of $O(|\mathcal{N}|^2 \log |\mathcal{N}|)$. Also, clustering based on label distributions does not always imply higher-quality rules, because it does not consider the interactions of the nonterminals on the left-hand side and the right-hand side in each synchronous rule.

3 Syntax-Label Clustering

As an alternative to using the similarity of probabilistic distributions as a criterion for syntax-label clustering, we propose a clustering method based on the maximum likelihood of the synchronous rules in a training data $\mathcal{D}$. We uses the idea of maximizing the Bayesian posterior probability $P(M|\mathcal{D})$ of the overall model structure M given data $\mathcal{D}$ (Stolcke and Omohundro, 1994). While their goal is to maximize the posterior

$$P(M|\mathcal{D}) \propto P(M)P(\mathcal{D}|M) \tag{3}$$

we omit the prior term $P(M)$ and directly maximize the $P(\mathcal{D}|M)$. A model M is a clustering structure[1] . The synchronous rule in the data $\mathcal{D}$ for SAMT with target-side syntax labels is represented as

$$X \rightarrow \langle a_1 Y^{(1)} a_2 Z^{(2)} a_3, b_1 Y^{(1)} b_2 Z^{(2)} b_3 \rangle \tag{4}$$

where a_1, a_2, a_3 and b_1, b_2, b_3 are the source- and target-side terminals, respectively X, Y, Z are nonterminal syntax labels, and the superscript number indicates alignment between the source- and target-side nonterminals. Using Equation (4) we maximize the posterior probability $P(\mathcal{D}|M)$ which we define as the probability of right-hand side given the syntax label X of the left-hand side rule in the training data as follows:

$$\sum_{X \rightarrow \langle \alpha, \beta, \sim \rangle \in \mathcal{D}} \log Pr(\langle \alpha, \beta, \sim \rangle | X) \tag{5}$$

For the sake of simplicity, we assume that the generative probability for each rule does not depend on the existence of terminal symbols and that the reordering in the target side may be ignored. Therefore, Equation (5) simplifies to

$$\sum_{X \rightarrow \langle a_1 Y^{(1)} a_2 Z^{(2)} a_3, b_1 Y^{(1)} b_2 Z^{(2)} b_3 \rangle} \log p(Y, Z|X) \tag{6}$$

[1] $P(M)$ is reflected by the number of clusters.

3.1 Optimization Criterion

The generative probability in each rule of the form of Equation (6) can be approximated by clustering nonterminal symbols as follows:

$$p(Y,Z|X) \approx p(Y|c(Y)) \cdot p(Z|c(Z)) \cdot p(c(Y),c(Z)|c(X)) \quad (7)$$

where we map a syntax label X to its equivalence cluster $c(X)$. This can be regarded as the clustering criterion usually used in a class-based n-gram language model (Brown et al., 1992). If each label on the right-hand side of a synchronous rule (4) is independent of each other, we can factor the joint model as follows:

$$p(Y,Z|X) \approx p(Y|c(Y)) \cdot p(Z|c(Z)) \cdot p(c(Y)|c(X)) p(c(Z)|c(X)) \quad (8)$$

We introduce the predictive idea of Uszkoreit and Brants (2008) to Equation (8), which doesn't condition on the clustered label $c(X)$, but directly on the syntax label X:

$$p(Y,Z|X) \approx p(Y|c(Y)) \cdot p(Z|c(Z)) \cdot p(c(Y)|X) \cdot p(c(Z)|X) \quad (9)$$

The objective in Equation (9) is represented using the frequency in the training data as

$$\frac{N(Y)}{N(c(Y))} \cdot \frac{N(X,c(Y))}{N(X)} \cdot \frac{N(Z)}{N(c(Z))} \cdot \frac{N(X,c(Z))}{N(X)} \quad (10)$$

where $N(X)$ and $N(c(X))$ denote the frequency[2] of X and $c(X)$, and $N(X,K)$ denotes the frequency of cluster K in the right-hand side of a synchronous rule whose left-hand side syntax label is X. By replacing the rule probabilities in Equation (9) with Equation (10) and plugging the result into Equation (6), our objective becomes

$$\begin{aligned} F(\mathcal{C}) &= \sum_{Y \in \mathcal{N}} N(Y) \cdot \log \frac{N(Y)}{N(c(Y))} + \sum_{X \in \mathcal{N}, K \in \mathcal{C}} N(X,K) \cdot \log \frac{N(X,K)}{N(X)} \\ &= \sum_{Y \in \mathcal{N}} N(Y) \cdot \log N(Y) - \sum_{Y \in \mathcal{N}} N(Y) \cdot \log N(c(Y)) + \sum_{X \in \mathcal{N}, K \in \mathcal{C}} N(X,K) \cdot \log N(X,K) - \sum_{X \in \mathcal{N}, K \in \mathcal{C}} N(X,K) \cdot \log N(X) \end{aligned} \quad (11)$$

[2] We use a fractional count (Chiang, 2007) which adds up to one as a frequency.

start with the initial mapping (label $X \rightarrow c(X)$)
compute objective function $F(\mathcal{C})$
for each label X do
remove label X from $c(X)$
for each cluster K do
move label X tentatively to cluster K
compute $F(\mathcal{C})$ for this exchange
move label X to cluster with maximum $F(\mathcal{C})$
do until the cluster mapping does not change

Table 1: Outline of syntax-label clustering method

where $\mathcal{C}$ denotes all clusters and $\mathcal{N}$ denotes all syntax labels. For Equation (11), the last summation is equivalent to the sum of the occurrences of all syntax labels, and canceled out by the first summation. K in the third summation considers clusters in a synchronous rule whose left-hand side label is X, and we let $ch(X)$ denote a set of those clusters. The second summation equals $\sum_{K \in C} N(K) \cdot \log N(K)$. As a result, Equation (11) simplifies to

$$F(\mathcal{C}) = \sum_{X \in \mathcal{N}, K \in ch(X)} N(X,K) \cdot \log N(X,K) - \sum_{K \in \mathcal{C}} N(K) \cdot \log N(K) \quad (12)$$

3.2 Exchange Clustering

We used an exchange clustering algorithm (Uszkoreit and Brants, 2008) which was proven to be very efficient in word clustering with a vocabulary of over 1 million words. The exchange clustering for words begins with the initial clustering of words and greedily exchanges words from one cluster to another such that an optimization criterion is maximized after the move. While agglomerative clustering requires recalculation for all pair-wise distances between words, exchange clustering only demands computing the difference of the objective for the word pair involved in a particular movement. We applied this exchange clustering to syntax-label clustering. Table 1 shows the outline. For initial clustering, we partitioned all the syntax labels into clusters according to the frequency of syntax labels in synchronous rules. If `remove` and `move` are as computationally intensive as computing the change in $F(\mathcal{C})$ in Equation (12), then the time complexity of `remove` and `move` is $O(K)$ (Martin et al., 1998), where K is the number of clusters. Since the `remove` procedure is called once for each label and, for a given label, the `move` procedure is called $K-1$ times

Data	Lang	Training			Development		Test	
		sent	src-tokens	tgt-tokens	sent	tgt-tokens	sent	tgt-tokens
IWSLT07	J to E	40 K	483 K	369 K	500	7.4 K	489	3.7 K
FBIS	C to E	302 K	2.7 M	3.4 M	1,664	47 K	919	30 K
NIST08		1 M	15 M	17 M				

Table 2: Data sets: The "sent" column indicates the number of sentences. The "src-tokens" and "tgt-tokens" columns indicate the number of words in the source- and the target-side sentences.

to find the maximum $F(\mathcal{C})$, the worst-time complexity for one iteration of the syntax-label clustering is $O(|\mathcal{N}|K^2)$. The exchange procedure is continued until the cluster mapping is stable or the number of iterations reaches a threshold value of 100.

4 Experiments

4.1 Data

We conducted experiments on Japanese-English (ja-en) and Chinese-English (zh-en) translation tasks. The ja-en data comes from IWSLT07 (Fordyce, 2007) in a spoken travel domain. The tuning set has seven English references and the test set has six English references. For zh-en data we prepared two kind of data. The one is extracted from FBIS[3], which is a collection of news articles. The other is 1 M sentences extracted rondomly from NIST Open MT 2008 task (NIST08). We use the NIST Open MT 2006 for tuning and the MT 2003 for testing. The tuning and test sets have four English references. Table 2 shows the details for each corpus. Each corpus is tokenized, put in lower-case, and sentences with over 40 tokens on either side are removed from the training data. We use KyTea (Neubig et al., 2011) to tokenize the Japanese data and Stanford Word Segmenter (Tseng et al., 2005) to tokenize the Chinese data. We parse the English data with the Berkeley parser (Petrov and Klein, 2007).

4.2 Experiment design

We did experiments with the SAMT (Zollmann and Venugopal, 2006) model with the Moses (Koehn et al., 2007). For the SAMT model, we conducted experiments with two label sets. One is extracted from the phrase structure parses and the other is extended with CCG[4]. We applied the proposed method (+*clustering*) and the baseline method (+*coarsening*), which uses the Hanneman label-collapsing algorithm described in Section 2, for syntax-label clustering to the SAMT models with *CCG*. The number of clusters for each clustering was set to 80. The language models were built using SRILM Toolkits (Stolcke, 2002). The language model with the IWSLT07 is a 5-gram model trained on the training data, and the language model with the FBIS and NIST08 is a 5-gram model trained on the Xinhua portion of English GigaWord. For word alignments, we used MGIZA++ (Gao and Vogel, 2008). To tune the weights for BLEU (Papineni et al., 2002), we used the n-best batch MIRA (Cherry and Foster, 2012).

[3]LDC2003E14

[4]Using the relax-parse with option SAMT 4 for IWSLT07 and FBIS and SAMT 2 for NIST08 in the Moses

Label set	Label	Rule	$\mathbf{F}(\mathcal{C})$	SD
parse	63	0.3 K	-	-
CCG	3,147	4.2 M	-	-
+ *coarsening*	80	2.4 M	-3.8 e+08	249
+ *clustering*	80	3.8 M	-7.2 e+07	73

Table 3: SAMT grammars on ja-en experiments

Label set	Label	Rule	$\mathbf{F}(\mathcal{C})$	SD
FBIS				
parse	70	2.1 M	-	-
CCG	5,460	60 M	-	-
+ *coarsening*	80	32 M	-1.5 e+10	526
+ *clustering*	80	38 M	-7.9 e+09	154
NIST08				
parse	70	12 M	-	-
CCG	7,328	120 M	-	-
+ *clustering*	80	100 M	-2.6 e+10	218

Table 4: SAMT grammars on zh-en experiments

5 Results and analysis

Tables 3 and 4 present the details of SAMT grammars with each label set learned by the experiments using the IWSLT07 (ja-en), FBIS and NIST08 (zh-en), which include the number of syntax labels and synchronous rules, the values of the objective ($F(\mathcal{C})$), and the standard deviation (SD) of the number of labels assigned to each cluster. For NIST08 we applied only the + *clustering* because the + *coarsening* needs a huge amount of computation time. Table 5 shows the differences between the BLEU score and the rule number for

each cluster number when using the IWSLT07 dataset.

Since the +*clustering* maximizes the likelihood of synchronous rules, it can introduce appropriate rules adapted to training data given a fixed number of clusters. For each experiment, SAMT grammars with the +*clustering* have a greater number of rules than with the +*coarsening* and, as shown in Table 5, the number of synchronous rules with +*clustering* increase with the number of clusters. For +*clustering* with eight clusters and +*coarsening* with 80 clusters, which have almost $2.4M$ rules, the BLEU score of +*clustering* with eight clusters is higher. Also, the SD of the number of labels, which indicates the balance of the number of labels among clusters, with +*clustering* is smaller than with +*coarsening*. These results suggest that +*clustering* maintain a large-scale variation of synchronous rules for high performance by balancing the number of labels in each cluster.

The number of synchronous rules grows as you progress from +*coarsening* to +*clustering* and finally to raw label with *CCG*. To confirm the effect of the number of rules, we measured the decoding time per sentence for translating the test set by taking the average of ten runs with FBIS corpus. +*coarsening* takes 0.14 s and +*clustering* takes 0.16 s while raw label with CCG takes 0.37s. Thus the increase in the number of synchronous rules adversely affects the decoding speed.

Table 6 presents the results for the experiments[5] using ja-en and zh-en with the BLEU metric. SAMT with *parse* have the lowest BLEU scores. It appears that the linguistic information of the raw syntax labels of the phrase structure parses is not enough to improve the translation performance. Hiero has the higher BLEU score than SAMT with *CCG* on zh-en. This is likely due to the low accuracy of the parses, on which SAMT relies while Hiero doesn't. SAMT with + *clustering* have the higher BLEU score than raw label with *CCG*. For SAMT with *CCG* using IWSLT07 and FBIS, though the statistical significance tests were not significant when $p < 0.05$, +*clustering* have the higher BLEU scores than +*coarsening*. For these results, the performance of +*clustering* is comparable to that of +*coarsening*. For the complexity of both clustering algorithm, though it is difficult to evaluate directly because the speed

[5]As another baseline, we also used Phrase-based SMT (Koehn et al., 2003) and Hiero (Chiang, 2007).

	+*clustering*				+*coarsening*
Cluster	**80**	**40**	**8**	**4**	**80**
BLEU	50.21	49.49	49.96	50.25	49.54
Rule	3.8 M	3.5 M	2.4 M	2.2 M	2.4 M

Table 5: BLEU score and rule number for each cluster number using IWSLT07

	ja-en		**zh-en**			
Model	***parse***	***CCG***	***parse***	***CCG***	***parse***	***CCG***
SAMT	42.58	48.77	23.66	26.97	24.67	27.28
+*coarsening*	-	49.54	-	27.12	-	-
+*clustering*	-	**50.21**	-	**27.47**	-	27.29
Hiero	48.91		**28.31**		**27.62**	
PB-SMT	49.14		26.88		26.71	

Table 6: BLEU scores on each experiments

depends on how each algorithm is implemented, +*clustering* is an order of magnitude faster than +*coarsening*. For the clustering experiment that groups 5460 raw labels with *CCG* into 80 clusters using FBIS corpus, +*coarsening* takes about 1 week whereas +*clustering* takes about 10 minutes.

6 Conclusion

In this paper, we propose syntax-label clustering for SAMT, which uses syntax-label information to generate syntactically correct translations. One of the problems of SAMT is the large grammar size when a CCG-style extended label set is used in the grammar, which make decoding slower. We cluster syntax labels with a very fast exchange algorithm in which the generative probabilities of synchronous rules are maximized. We demonstrate the effectiveness of the proposed method by using it to translate Japanese-English and Chinese-English tasks and measuring the decoding speed, the accuracy and the clustering speed. Future work involves improving the optimization criterion. We expect to make a new objective that includes the terminal symbols and the reordering of nonterminal symbols that were ignored in this work. Another interesting direction is to determine the appropriate number of clusters for each corpus and the initialization method for clustering.

Acknowledgments

We thank the anonymous reviewers for their suggestions and helpful comments on the early version of this paper.

References

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, pages 201–228, June.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

Cameron Shaw Fordyce. 2007. Overview of the 4th international workshop on spoken language translation iwslt 2007 evaluation campaign. In *In Proceedings of IWSLT 2007*, pages 1–12, Trento, Italy, October.

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 the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June. Association for Computational Linguistics.

Greg Hanneman and Alon Lavie. 2013. Improving syntax-augmented machine translation by coarsening the label set. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 288–297, Atlanta, Georgia, June. Association for Computational Linguistics.

Bryant Huang and Kevin Knight. 2006. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 240–247, New York City, USA, June. Association for Computational Linguistics.

Phillip Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *In Proceedings of HLT-NAACL*, pages 48–54, Edmonton, Canada, May/July.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Sven Martin, Jorg Liermann, and Hermann Ney. 1998. Algorithms for bigram and trigram word clustering. In *Speech Communication*, pages 19–37.

Markos Mylonakis and Khalil Sima'an. 2011. Learning hierarchical translation structure with linguistic annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–652, Portland, Oregon, USA, June. Association for Computational Linguistics.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Mark Steedman. 2000. *The syntactic process*, volume 27. MIT Press.

Andreas Stolcke and Stephen Omohundro. 1994. Inducing probabilistic grammars by bayesian model

merging. In R. C. Carrasco and J. Oncina, editors, *Grammatical Inference and Applications (ICGI-94)*, pages 106–118. Berlin, Heidelberg.

Andreas Stolcke. 2002. Srilm an extensible language modeling toolkit. In *In Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. *In Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171. Jeju Island, Korea.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *Proceedings of ACL-08: HLT*, pages 755–762, Columbus, Ohio, June. Association for Computational Linguistics.

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