4-4 Direct Use of Syntactic Information for Machine Translation System Combination

WATANABE Taro

The state-of-the-art system combination method for machine translation (MT) is based on confusion networks constructed by aligning hypotheses with regard to word similarities. We introduce a novel system combination framework in which hypotheses are encoded as a confusion forest, a packed forest representing alternative trees. The forest is generated using syntactic consensus among parsed hypotheses: First, MT outputs are parsed. Second, a context free grammar is learned by extracting a set of rules that constitute the parse trees. Third, a packed forest is generated starting from the root symbol of the extracted grammar through non-terminal rewriting. The new hypothesis is produced by searching the best derivation in the forest. Experimental results on the WMT10 system combination shared task yield comparable performance to the conventional confusion network based method with smaller space.

Keywords

Statistical machine translation, System output combination, Phrase structure parse tree, Forest

1 Introduction

System combination techniques take the advantages of consensus among multiple systems and have been widely used in fields, such as speech recognition [1][2] or parsing [3]. One of the state-of-the-art system combination methods for MT is based on confusion networks, which are compact graph-based structures representing multiple hypotheses [4].

Confusion networks are constructed based on string similarity information. First, one skeleton or backbone sentence is selected. Then, other hypotheses are aligned against the skeleton, forming a lattice with each arc representing alternative word candidates. The alignment method is either model-based [5][6] in which a statistical word aligner is used to compute hypothesis alignment, or edit-based [7][8] in which alignment is measured by an evaluation metric, such as translation error rate (TER) [9]. The new translation hypothesis is generated by selecting the best path through the network.

We present a novel method for system combination which exploits the syntactic similarity of system outputs. Instead of constructing a string-based confusion network, we generate a packed forest [10][11] which encodes exponentially many parse trees in a polynomial space. The packed forest, or *confusion forest*, is constructed by merging the MT outputs with regard to their syntactic consensus. We employ a grammar-based method to generate the confusion forest: First, system outputs are parsed. Second, a set of rules are extracted from the parse trees. Third, a packed forest is generated using a variant of Earley's algorithm [12] starting from the unique root symbol. New hypotheses are selected by searching the best derivation in the forest. The grammar, a set of rules, is limited to those found in the parse trees. Spurious ambiguity during the generation step is further reduced by encoding the tree local contextual information in each non-terminal symbol, such as parent and sibling labels, using the state representation in Earley's algorithm.

Experiments were carried out for the system combination task of the fifth workshop on statistical machine translation (WMT10) in four directions, {Czech, French, German, Spanish}-to-English [13], and we found comparable performance to the conventional confusion network based system combination in two language pairs, and statistically significant improvements in the others.

2 Confusion network

The system combination framework based on confusion network starts from computing pairwise alignment between hypotheses by taking one hypothesis as a reference. [5] employs a model based approach in which a statistical word aligner, such as GIZA++ [14], is used to align the hypotheses. [8] introduced TER [9] to measure the edit-based alignment.

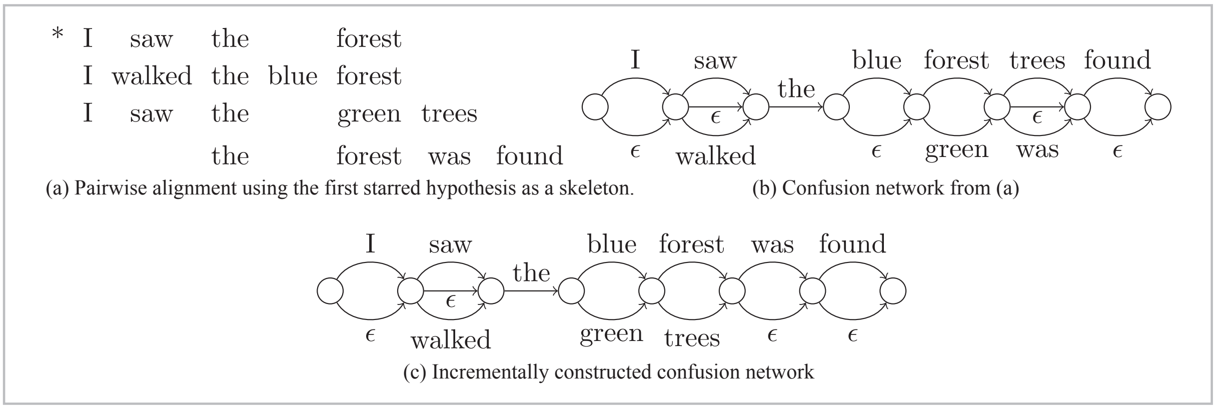
Then, one hypothesis is selected, for example by employing a minimum Bayes risk criterion [8], as a skeleton, or a backbone, which serves as a building block for aligning the rest of the hypotheses. Other hypotheses are aligned against the skeleton using the pairwise alignment. Figure 1(b) illustrates an example of a confusion network constructed from the four hypotheses in Fig. 1(a), assuming the first hypothesis is selected as our skeleton. The network consists of several arcs, each of which represents an alternative word at that position, including the empty symbol, ϵ.

This pairwise alignment strategy is prone to spurious insertions and repetitions due to alignment errors such as in Fig. 1(a) in which "green" in the third hypothesis is aligned with "forest" in the skeleton. [15] introduces an incremental method so that hypotheses are aligned incrementally to the growing confusion network, not only the skeleton hypothesis. In our example, "green trees" is aligned with "blue forest" in Fig. 1(c).

The confusion network construction is largely influenced by the skeleton selection, which determines the global word reordering of a new hypothesis. For example, the last hypothesis in Fig. 1(a) has a passive voice grammatical construction while the others are active voice. This large grammatical difference may produce a longer sentence with spuriously inserted words, as in "I saw the blue trees was found" in Fig. 1(c). [16] partially resolved the problem by constructing a large network in which each hypothesis was treated as a skeleton and the multiple networks were merged into a single network.

3 Confusion forest

The confusion network approach to system combination encodes multiple hypotheses into a compact lattice structure by using word-level consensus. Likewise, we propose to encode multiple hypotheses into a confusion forest, which is a packed forest which represents mul-



(a) Pairwise alignment using the first starred hypothesis as a skeleton.

(b) Confusion network from (a)

(c) Incrementally constructed confusion network

Fig.1 *An example confusion network construction*

tiple parse trees in a polynomial space [10][11]. Syntactic consensus is realized by sharing tree fragments among parse trees. The forest is represented as a hypergraph which is exploited in parsing [17][18] and machine translation [19][20].

More formally, a hypergraph is a pair $\langle V, E\rangle$ where V is the set of nodes and E is the set of hyperedges. Each node in V is represented as $X^{@p}$ where $\mathrm{X} \in N$ is a non-terminal symbol and p is an address [21] that encapsulates each node id relative to its parent. The root node is given the address ϵ and the address of the first child of node p is given $p.1$. Each hyperedge $e \in E$ is represented as a pair $\langle head(e), tails(e)\rangle$ where $head(e) \in V$ is a head node and $tails(e) \in V^*$ is a list of tail nodes, corresponding to the left-hand side and the right-hand side of an instance of a rule in a CFG, respectively. Figure 2 presents an example packed forest for the parsed hypotheses in Fig. 1(a). For example, $\mathrm{VP}^{@2}$ has two hyperedges, $\langle \mathrm{VP}^{@2}, (\mathrm{VBD}^{@3}, \mathrm{VP}^{@4})\rangle$ and $\langle \mathrm{VP}^{@2}, (\mathrm{VBD}^{@2.1}, \mathrm{NP}^{@2.2})\rangle$, leading to different derivations where the former takes the grammatical construction in passive voice while the latter in active voice.

Given system outputs, we employ the following grammar based approach for constructing a confusion forest: First, MT outputs are parsed. Second, a grammar is learned by treating each hyperedge as an instance of a CFG rule. Third, a forest is generated from the unique root symbol of the extracted grammar through non-terminal rewriting.

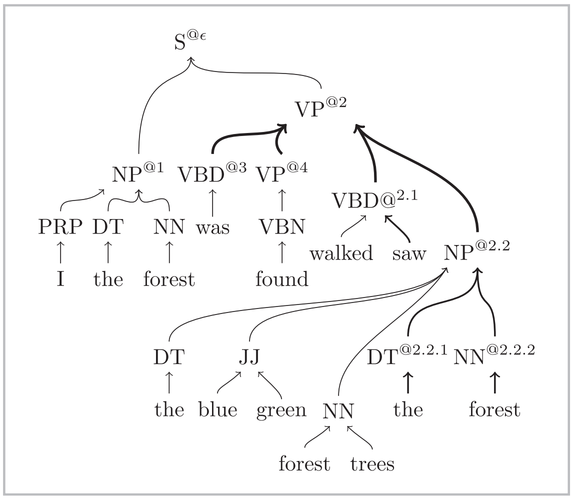


Fig.2 An example packed forest representing hypotheses in Fig. 1(a)

3.1 Rule extraction

During the rule extraction procedure, we reduce spurious ambiguities o the extracted grammar by encoding the original tree structures in each node. First, horizontal Markovization [23] encodes sibling labels in each non-terminals. For instance, Fig. 3(a) presents a parse tree for a system output "I saw the forest." In Figure 3(b), the sub-tree rooted at the node $\mathrm{VP}^{@2}$ in Fig. 3(a) is annotated by our labeling method. For example, $\mathrm{NP}^{@2.2}$ is combined with its sibling $\mathrm{VBD}^{@2.1}$ with ● representing the original label positions.

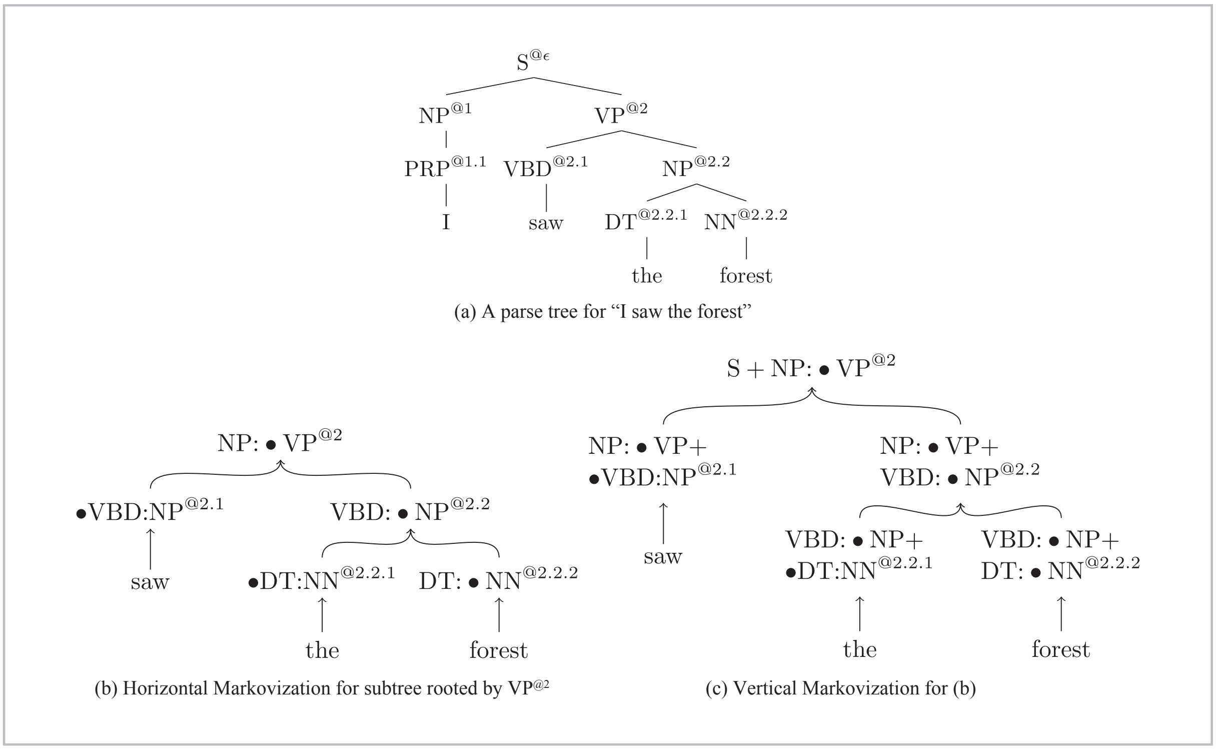
Next, vertical Markovization [23] combines parent labels. In Figure 3(c), the node at @2.2 is combined with its parent node @2, yielding the new label (NP: ● VP + VBD: ● NP). After the label annotation, we extract a grammar by treating each hyperedge as a rule.

The context represented in each node is further limited by the vertical and horizontal Markovization [23]. We define the vertical order v in which the label is limited to memorize only v previous parents. Likewise, we introduce the horizontal order h which limits the number of sibling labels memorized on the left and the right of the dotted label.

No limits in the horizontal and vertical Markovization orders implies memorizing of all the original tree structures and yields a confusion forest representing the union of parse trees through the grammar collection and the generation processes. More relaxed horizontal orders allow more reordering of subtrees in a confusion forest by discarding the sibling context. Likewise, constraining the vertical order generates a deeper forest by ignoring the sequence of symbols leading to a particular node.

3.2 Forest generation

Given the extracted grammar, we apply a variant of Earley's algorithm [12] which can



(a) A parse tree for "I saw the forest"

(b) Horizontal Markovization for subtree rooted by VP@2

(c) Vertical Markovization for (b)

Fig.3 *Label annotation by horizontal/vertical Markovization*

generate strings in a left-to-right manner from the unique root symbol, TOP. Figure 4 presents the deductive inference rules [22] for our generation algorithm. We use capital letters X ∈ *N* to denote non-terminals and $x \in T$ for terminals. Lowercase Greek letters α, β and γ are strings of terminals and non-terminals $(T \cup N)^*$. u and v are weights associated with each item.

The major difference compared to Earley's parsing algorithm is that we ignore the terminal span information each non-terminal covers and keep track of the height of derivations by h. The scanning step will always succeed by moving the dot to the right. Combined with the prediction and completion steps, our algorithm may potentially generate a spuriously deep forest. Thus, the height of the forest is constrained in the prediction step not to exceed H, which is empirically set to 1.5 times the maximum height of the parsed system outputs.

Initialization:

$$\frac{}{[\mathrm{TOP} \rightarrow \bullet \mathrm{S}, 0] : \bar{1}}$$

Scan:

$$\frac{[\mathrm{X} \rightarrow \alpha \bullet x\beta, h] : u}{[\mathrm{X} \rightarrow \alpha x \bullet \beta, h] : u}$$

Predict:

$$\frac{[\mathrm{X} \rightarrow \alpha \bullet \mathrm{Y}\beta, h]}{[\mathrm{Y} \rightarrow \bullet\gamma, h+1] : u} \quad \mathrm{Y} \xrightarrow{u} \gamma \in \mathcal{G}, h < H$$

Complete:

$$\frac{[\mathrm{X} \rightarrow \alpha \bullet \mathrm{Y}\beta, h] : u \quad [\mathrm{Y} \rightarrow \gamma\bullet, h+1] : v}{[\mathrm{X} \rightarrow \alpha\mathrm{Y} \bullet \beta, h] : u \otimes v}$$

Goal:

$$[\mathrm{TOP} \rightarrow \mathrm{S}\bullet, 0]$$

Fig.4 *The deductive system for Earley's generation algorithm*

3.3 Forest rescoring

From the packed forest F, new k-best derivations are extracted from all possible derivations D by efficient forest-based algorithms for k-best parsing [18]. We use a linear combination of features as our objective function to seek for the best derivation $\hat{d}$:

$$\hat{d} = \operatorname*{arg\,max}_{d \in D} \boldsymbol{w}^{\top} \cdot \boldsymbol{h}(d, F) \tag{1}$$

where $\boldsymbol{h}$ (d, F) is a set of feature functions scaled by weight vector $\boldsymbol{w}$. We use cube-pruning [19][20] to approximately intersect with non-local features, such as n-gram language models. Then, k-best derivations are extracted from the rescored forest using algorithm 3 of [18].

4 Experiments

4.1 Setup

We ran our experiments for the WMT10 system combination task usinge four language pairs, {Czech, French, German, Spanish}-to-English [13]. The data is summarized in Table 1. The system outputs are retokenized to match the Penn-treebank standard, parsed by the Stanford Parser [23], and lower-cased.

We implemented our confusion forest system combination using an in-house developed hypergraphbased toolkit *cicada* which is motivated by generic weighted logic programming [24], originally developed for a synchronous-CFG based machine translation system [19]. Input to our system is a collection of hypergraphs, a set of parsed hypotheses, from which rules are extracted and a new forest is generated as described in Section **3**. Our baseline, also implemented in *cicada*, is a confusion network-based system combination method (See in Section **2**) which incrementally aligns hypotheses to the growing network using TER [15] and merges multiple networks into a large single network. After performing epsilon removal, the network is transformed into a forest by parsing with monotone rules of S → X, S → S X and X → x. k-best translations are extracted from the forest using the forest-based algorithms in Subsection **3.3**.

Table 1 *WMT10 system combination tuning/testing data*

		cz-en	de-en	es-en	fr-en
# of systems		6	16	8	14
avg. words	tune	10.6K	10.9K	10.9K	11.0K
	test	50.5K	52.1K	52.1K	52.4K
sentences	tune	455			
	test	2,034			

4.2 Features

The feature weight vector w in Equation 1 is tuned by MERT over hypergraphs [25].

We use three lower-cased 5-gram language models $h^i_{lm}(d)$: English Gigaword Fourth edition[*1], the English side of French-English 10^9 corpus and the news commentary English data[*2]. The count based features $h_t(d)$ and $h_e(d)$ count the number of terminals and the number of hyperedges in d, respectively. We employ M confidence measures h^m_s (d) for M systems, which basically count the number of rules used in d originally extracted from mth system hypothesis [26].

Following [27], BLEU [28] correlations are also incorporated in our system combination. Given M system outputs $\boldsymbol{e}_1 \ldots \boldsymbol{e}_M$, M BLEU scores are computed for d using each of the system outputs $\boldsymbol{e}_m$ as a reference

$$h_b^m(d) = BP(\boldsymbol{e}, \boldsymbol{e}_m) \cdot \exp\left(\frac{1}{4}\sum_{n=1}^{4} \log \rho_n(\boldsymbol{e}, \boldsymbol{e}_m)\right)$$

where $\boldsymbol{e}$ = yield(d) is a terminal yield of d, $BP(\cdot)$ and $\rho_n(\cdot)$ respectively denote brevity penalty and N-gram precision. Here, we use approximated unclipped N-gram counts [29] for computing $\rho_n(\cdot)$ with a compact state representation [30].

Our baseline confusion network system has an additional penalty feature, $h_p(m)$, which is the total edits required to construct a confusion network using the mth system hypothesis as a skeleton, normalized by the number of nodes in the network [16].

4.3 Results

Table 2 compares our confusion forest approach (CF) with different orders, a confusion network (CN) and max/min systems measured by BLEU [28]. We vary the horizontal orders, $h = 1, 2, \infty$ with vertical orders of $v = 3, 4, \infty$. Systems without statistically significant differ-

[*1] LDC catalog No. LDC2009T13

[*2] Those data are available from http://www.statmt.org/wmt10/.

ences from the best result ($p < 0.05$) are indicated by bold face. Setting $v = \infty$ and $h = \infty$ achieves comparable performance to CN. Our best results in three languages come from setting $v = \infty$ and $h = 2$, which favors little reordering of phrasal structures. In general, lower horizontal and vertical order leads to lower BLEU.

Introducing new tree fragments to confusion forests leads to new phrasal translations with enlarged forests, as presented in Table 3, measured by the average number of hyperedges[*3]. The larger potentials do not imply better translations, probably due to the larger search space with increased search errors. We also conjecture that syntactic variations were not captured by the N-gram like string-based features in Chapter **4-2**, therefore resulting in BLEU loss, which will be investigated in future work.

Table 2 *Translation results in lower-case BLEU. CN for confusion network and CF for confusion forest with different vertical (v) and horizontal (h) Markovization order*

language	cz-en	de-en	es-en	fr-en
system min	14.09	15.62	21.79	16.79
max	23.44	24.10	29.97	**29.17**
CN	23.70	24.09	**30.45**	**29.15**
CF$_{v=\infty, h=\infty}$	**24.13**	24.18	**30.41**	**29.57**
CF$_{v=\infty, h=2}$	**24.14**	**24.58**	**30.52**	28.84
CF$_{v=\infty, h=1}$	**24.01**	23.91	**30.46**	**29.32**
CF$_{v=4, h=\infty}$	**23.93**	23.57	29.88	28.71
CF$_{v=4, h=2}$	**23.82**	22.68	29.92	28.83
CF$_{v=4, h=1}$	**23.77**	21.42	30.10	28.32
CF$_{v=3, h=\infty}$	23.38	23.34	29.81	27.34
CF$_{v=3, h=2}$	23.30	23.95	30.02	28.19
CF$_{v=3, h=1}$	23.23	21.43	29.27	26.53

Table 3 *Hypegraph size measured by the average number of hyperedges (h = 1 for CF). "lattice" is the average number of edges in the original CN*

lang	cz-en	de-en	es-en	fr-en
CN	2,222.68	47,231.20	2,932.24	11,969.40
lattice	1,723.91	41,403.90	2,330.04	10,119.10
CF$_{v=\infty}$	230.08	540.03	262.30	386.79
CF$_{v=4}$	254.45	651.10	302.01	477.51
CF$_{v=3}$	286.01	802.79	349.21	575.17

Table 3 also shows that CN produces a forest that is an order of magnitude larger than those created by CFs. Although we cannot directly relate the runtime and the number of hyperedges in CN and CFs, since the shape of the forests are different, CN requires more space to encode the hypotheses than those by CFs.

5 Conclusion

We presented a confusion forest based method for system combination in which system outputs are merged into a packed forest using their syntactic similarity. The forest construction is treated as a generation from a CFG compiled from the parsed outputs. Our experiments indicate comparable performance to a strong confusion network baseline with smaller space, and statistically significant gains in some language pairs.

To our knowledge, this is the first work to directly introduce syntactic consensus to system combination by encoding multiple system outputs into a single forest structure. We believe that the confusion forest based approach to system combination has future exploration potential. For instance, we did not employ syntactic features in Subsection **4.2** which would be helpful in discriminating hypotheses in larger forests. We would also like to analyze the trade-offs, if any, between parsing errors and confusion forest constructions by controlling the parsing qualities. As an alternative to the grammar-based forest generation, we are investigating an edit distance measure for tree alignment, such as tree edit distance [31] which basically computes insertion/deletion/replacement of nodes in trees.

[*3] We measure the hypergraph size before intersecting with non-local features, like n-gram language models.

References

1 J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover),” In Proc. of ASRU, pp. 347–354, Dec. 1997.

2 Lidia Mangu, Eric Brill, and Andreas Stolcke, “Finding consensus in speech recognition: word error minimization and other applications of confusion networks,” Computer Speech & Language, 14(4): 373–400, 2000.

3 John C. Henderson and Eric Brill, “Exploiting diversity in natural language processing: Combining parsers,” In IN PROCEEDINGS OF THE FOURTH CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, pp. 187–194, 1999.

4 Srinivas Bangalore, German Bordel, and Giuseppe Riccardi, “Computing consensus translation from multiple machine translation systems,” In Proc. of ASRU, pp. 351–354, 2001.

5 Evgeny Matusov, Nicola Ueffing, and Hermann Ney, “Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment,” In Proc. of EACL, pp. 33–40, 2006.

6 Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore, “Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems,” In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 98–107, Honolulu, Hawaii, Oct. 2008. Association for Computational Linguistics.

7 Shyamsundar Jayaraman and Alon Lavie, “Multi-engine machine translation guided by explicit word matching,” In Proceedings of the ACL 2005 on Interactive poster and demonstration sessions, ACL ’05, pp. 101–104, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

8 K. C. Sim, W. J. Byrne, M. J. F. Gales, H. Sahbi, and P. C. Woodland, “Consensus network decoding for statistical machine translation system combination,” In Proc. of ICASSP, Vol. 4, pp. IV-105–IV-108, April 2007.

9 Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul, “A study of translation edit rate with targeted human annotation,” In Proc. of AMTA, pp. 223–231, 2006.

10 Sylvie Billott and Bernard Lang, “The structure of shared forests in ambiguous parsing,” In Proc. of ACL, pp. 143–151, June 1989.

11 Haitao Mi, Liang Huang, and Qun Liu, “Forest-based translation,” In Proceedings of ACL-08: HLT, pp. 192–199, June 2008.

12 Jay Earley, “An efficient context-free parsing algorithm,” Communications of the Association for Computing Machinery, 13: 94–102, Feb. 1970.

13 Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan, “Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation,” In Proc. of WMT, pp. 17–53, July 2010.

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

15 Antti-Veikko Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz, “Incremental hypothesis alignment for building confusion networks with application to machine translation system combination,” In Proc. of WMT, pp. 183–186, June 2008.

16 Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz, “Improved word-level system combination for machine translation,” In Proc. of ACL, pp. 312–319, June 2007.

17 Dan Klein and Christopher D. Manning, “Parsing and hypergraphs,” In Proc. of IWPT, pp. 123–134, 2001.

18 Stuart M. Shieber, Yves Schabes, and Fernando C. N. Pereira, “Principles and implementation of deductive parsing,” Journal of Logic Programming, 24(1-2): 3–36, July-Aug. 1995.

19 Dan Klein and Christopher D. Manning, “Accurate unlexicalized parsing,” In Proc. of ACL, pp. 423–430, July 2003.

20 Joshua Goodman, “Semiring parsing,” Computational Linguistics, 25: 573–605, Dec. 1999.

21 Liang Huang and David Chiang, “Better k-best parsing,” In Proc. of IWPT, pp. 53–64, Oct. 2005.

22 David Chiang, “Hierarchical phrase-based translation,” Computational Linguistics, 33(2): 201–228, 2007.

23 Liang Huang and David Chiang, “Forest rescoring: Faster decoding with integrated language models,” In Proc. of ACL, pp. 144–151, June 2007.

24 Adam Lopez, “Translation as weighted deduction,” In Proc. of EMNLP, pp. 532–540, March 2009.

25 Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och, “Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices,” In Proc. of ACL/IJCNLP, pp. 163–171, Aug. 2009.

26 Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr, “Combining outputs from multiple machine translation systems,” In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 228–235, Rochester, New York, April 2007. Association for Computational Linguistics.

27 Wolfgang Macherey and Franz J. Och, “An empirical study on computing consensus translations from multiple machine translation systems,” In Proc. of EMNLP-CoNLL, pp. 986–995, June 2007.

28 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, “Bleu: a method for automatic evaluation of machine translation,” In Proc. of ACL, pp. 311–318, July 2002.

29 Zhifei Li and Sanjeev Khudanpur, “Efficient extraction of oracle-best translations from hypergraphs,” In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 9–12, Boulder, Colorado, June 2009. Association for Computational Linguistics.

30 Markus Dreyer, Keith Hall, and Sanjeev Khudanpur, “Comparing reordering constraints for smt using efficient bleu oracle computation,” In Proceedings of SSST, NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation, pp. 103–110, Rochester, New York, April 2007. Association for Computational Linguistics.

31 Philip Bille, “A survey on tree edit distance and related problems,” Theor. Comput. Sci., 337: 217–239, June 2005.

(Accepted June 14, 2012)

WATANABE Taro, *Ph.D.*

Senior Researcher, Multilingual Translation Laboratory, Universal Communication Research Institute

Machine Learning, Machine Translation, Natural Language Processing