NTT-NAIST SMT Systems for IWSLT 2013

Katsuhito Sudoh[*], *Graham Neubig*[†], *Kevin Duh*[†], *Hajime Tsukada*[*]

[*] NTT Communication Science Laboratories, Kyoto, Japan
[†] Nara Institute of Science and Technology, Nara, Japan
sudoh.katsuhito@lab.ntt.co.jp

Abstract

This paper presents NTT-NAIST SMT systems for English-German and German-English MT tasks of the IWSLT 2013 evaluation campaign. The systems are based on generalized minimum Bayes risk system combination of three SMT systems: forest-to-string, hierarchical phrase-based, phrase-based with pre-ordering. Individual SMT systems include data selection for domain adaptation, rescoring using recurrent neural net language models, interpolated language models, and compound word splitting (only for German-English).

1. Introduction

Spoken language is a very important and also challenging target for machine translation. MT tasks in the IWSLT evaluation campaign [1] focus on translating subtitles of speech from TED Talks. These subtitles are clean transcriptions without disfluencies that sometimes appeared in original talks. These talks can be expected to be similar to written texts that have been tackled in recent machine translation studies, as the talks are logically and syntactically well-organized compared to conversational speeches.

In our system this year, we focused on applying syntax-oriented translation technologies for statistical machine translation (SMT) such as forest-to-string translation and syntax-based pre-ordering. We also made several improvements to the base SMT models: domain adaptation by training data selection among different data sources; rescoring using recurrent neural network language models (RNNLMs); n-gram language model interpolation; compound word splitting for German compounds; and system combination of different types of SMT systems based on generalized minimum Bayes risk (GMBR) framework. This paper presents details of our systems and reports the results in German-English and English-German MT tasks in the evaluation campaign.

2. Translation Methods

The main feature of our system for this evaluation is that we perform translation using three different translation models and combine the results through system combination. Each of the three methods is described briefly below.

2.1. Phrase-based Machine Translation

Phrase-based machine translation (PBMT; [2]) models the translation process by splitting the source sentence into phrases, translating the phrases into target phrases, and reordering the phrases into the target language order. PBMT is currently the most widely used method in SMT as it is robust, does not require the availability of linguistic analysis tools, and achieves high accuracy, particularly for languages with similar syntactic structure.

2.2. Hierarchical Phrase-based Machine Translation

Hierarchical phrase-based machine translation (Hiero; [3]) expands the class of translation rules that can be used in phrase-based machine translation by further allowing rules with gaps that can be filled in a hierarchical fashion. Hiero is generally considered to be more accurate than PBMT on language pairs that are less monotonic, but also requires a significantly larger amount of memory and decoding time. As the German-English pair has a significant amount of reordering, particularly with movement of verbs, we can expect that Hiero will be able to handle these reorderings more appropriately in some cases.

2.3. Forest-to-string Machine Translation

Tree-to-string machine translation (T2S; [4]) performs translation by first syntactically parsing the source sentence, then translating from sub-structures of the parse to a string in the target language. Forest-to-string machine translation (F2S; [5]) generalizes this framework, making it possible to not only translate the single one-best syntactic parse, but a packed forest that encodes many possible parses, helping to pass along some of the ambiguity of parsing to be resolved during translation. While there are a number of proposed methods for incorporating source-side syntax into the translation process, here we use a method based on tree-to-string transducers [6].

Syntax-driven methods such as T2S and F2S are particularly useful for language pairs with extremely large amounts of reordering, as the syntactic parse can help guide the accurate re-ordering of entire phrases or clauses. On the other hand, these methods are highly dependent on parsing accuracy, and also have limits on the rules that can be extracted,

and are somewhat less robust than the previous two methods.

3. SMT Technologies

3.1. Training data selection

The target TED domain is different in both style and vocabulary from many of the other bitexts, e.g. Europarl, CommonCrawl (which we collectively call "general-domain" data[1]). To address this domain adaption problem, we performed adaptation training data selection using the method of [7].[2] The intuition is to select general-domain sentences that are similar to in-domain text, while being dis-similar to the average general-domain text.

To do so, one defines the score of an general-domain sentence pair (e, f) as [8]:

$$[IN_E(e) - GEN_E(e)] + [IN_F(f) - GEN_F(f)] \quad (1)$$

where $IN_E(e)$ is the *length-normalized* cross-entropy of e on the English in-domain LM. $GEN_E(e)$ is the length-normalized cross-entropy of e on the English general-domain LM, which is built from a sub-sample of the general-domain text. By taking a sub-sample (same size as the target-domain data), we reduce training time and avoid training and testing language models on the same general-domain data. Similarly, $IN_F(f)$ and $GEN_F(f)$ are the cross-entropies of f on Foreign-side LM. Finally, sentence pairs are ranked according to Eq. 1 and those with scores lower than some empirically-chosen threshold e.g. we choose this threshold by comparing BLEU on the dev set) are added together with the in-domain bitext for translation model training. Here, the LMs are Recurrent Neural Network Language Models (RNNLMs), which have been shown to outperform n-gram LMs in this problem [7].

3.2. Syntactic Rule-based Pre-ordering

Preordering is a method that attempts to first re-order the source sentence into a word order that is closer to the target. As German and English have significantly different word order, we can imagine that this will help our accuracy for this language pair.

3.2.1. German-to-English

We applied the clause restructuring method of Collins et al. [9] for German pre-ordering. The method is mainly based on moving German verbs in the end of clause structures towards the beginning of the clause. We re-implemented the method for German parse trees created using the Berkeley parser trained on TIGER corpus. We ignored some additional syntactic information such as subject markers and heads implemented in the original method of [9], because we used a different syntactic parser that did not provide this information.

3.2.2. English-to-German

We also tried to apply pre-ordering to English-to-German. We essentially did this by reversing the Collins German-to-English rules by moving some words towards the end of their siblings based on their part-of-speech tags as follows:

- in main clauses, `VB` words were moved,
- in subordinate clauses, `MD`, `VBP`, `VBD`, `VBZ` words were moved.

3.3. RNNLM Rescoring

Continuous-space language models using neural networks have attracted recent attention as a method to improve the fluency of output of MT or speech recognition. In our system, we used the recurrent neural network language model (RNNLM) of [10].[3] This model uses a continuous space representation over the language model state that is remembered throughout the entire sentence, and thus has the potential to ensure the global coherence of the sentence to the greater extent than simpler n-gram language models.

We incorporate the RNNLM probabilities through rescoring. For each system, we first output a 10,000-best list, then calculate the RNNLM log probabilities and add them as an additional feature to each translation hypothesis. We then re-run a single MERT optimization to find ideal weights for this new feature, and then extract the 1-best result from the 10,000-best list for the test set according to these new weights. The parameters for RNNLM training are tuned on the dev set to maximize perplexity, resulting in 300 hidden layers, 300 classes, and 4 steps of back-propogation through time.

3.4. German compound word splitting

German compound words present sparsity challenges for machine translation. To address this, we split German words following the general approach of [11]. The idea is to split a word if the geometric average of its subword frequencies is larger than whole word frequency. In our implementation, for each word, we searched for all possible decompositions into two sub-words, considering the possibility of deleting common German fillers "e", "es", and "s" (as in "Arbeit+s+tier"). For simplicity, we did not experiment with splitting into three or more sub-words as done in the `compound-splitter.perl` script distributed with the Moses package. The unigram frequencies for the subwords and whole word is computed from the German part of the bitext. This simple algorithm is especially useful for handling out-of-vocabulary and rare compound words that have high frequency sub-words in the training data. For the F2S sys-

[1] To give a sense of the domain difference, a 4-gram LM trained with Kneser-Ney smoothing on TED data gives a perplexity of 355 on the general domain data, compared to a perplexity of 99 on held-out TED data.

[2] Code/scripts available at http://cl.naist.jp/∼kevinduh/a/acl2013

[3] http://www.fit.vutbr.cz/~imikolov/rnnlm/

tem, sub-words are given the same POS tag as the original whole word.

In the evaluation campaign, we performed compound splitting only in the German-to-English task. We do not attempt to split German words for the English-to-German task, since it is non-trivial to handle recombination of German split words after reordering and translation.

3.5. GMBR system combination

We used a system combination method based on Generalized Minimum Bayes Risk optimization [12], which has been successfully applied to different types of SMT systems for patent translation [13]. Note that our system combination only picks one hypothesis from an N-best list and does not generate a new hypothesis by mixing partial hypotheses among the N-best.

3.5.1. Theory

Minimum Bayes Risk (MBR) is a decision rule to choose hypotheses that minimize the expected loss. In the task of SMT from a French sentence (f) to an English sentence (e), the MBR decision rule on $\delta(f) \rightarrow e'$ with the loss function L over the possible space of sentence pairs ($p(e, f)$) is denoted as:

$$\underset{\delta(f)}{\operatorname{argmin}} \sum_{e} L(\delta(f)|e)p(e|f) \tag{2}$$

In practice, we approximate this using N-best list $N(f)$ for the input f.

$$\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e)p(e|f) \tag{3}$$

Although MBR works effectively for re-ranking single system hypotheses, it is challenging for system combination because the estimated $p(e|f)$ from different systems cannot be reliably compared. One practical solution is to use uniform $p(e|f)$ but this does not achieve Bayes Risk. GMBR corrects by parameterizing the loss function as a linear combination of sub-components using parameter $\boldsymbol{\theta}$:

$$L(e'|e;\boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{4}$$

For example, suppose the desired loss function is "1.0−BLEU". Then the sub-components could be "1.0−precision(n-gram) $(1 \leq n \leq 4)$" and "brevity penalty".

Assuming uniform $p(e|f)$, the MBR decision rule can be denoted as:

$$\begin{aligned} &\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|} \\ &= \underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \end{aligned} \tag{5}$$

To ensure that the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$, we use a small development set to tune the parameter $\boldsymbol{\theta}$ as follows. For any two hypotheses e_1, e_2, and a reference translation e_r (possibly not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{6}$$

so that GMBR would select the hypothesis achieving lower loss. Conversely if e_2 is a better hypothesis, then we want opposite relation:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) > \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{7}$$

Thus, we directly compute the true loss using a development set and ensure that our GMBR decision rule minimizes this loss.

3.5.2. Implementation

We implement GMBR for SMT system combination as follows.

First we run SMT decoders to obtain N-best lists for all sentences in the development set, and extract all pairs of hypotheses where a difference exists in the true loss. Then we optimize $\boldsymbol{\theta}$ in a formulation similar to a Ranking SVM [14]. The pair-wise nature of Eqs. 6 and 7 makes the problem amendable to solutions in "learning to rank" literature [15]. We used BLEU as the objective function and the sub-components of BLEU as features (system identity feature was not used). There is one regularization hyperparameter for the Ranking SVM, which we set by cross-validation over the development set (`dev2010`).

3.6. What Didn't Work Immediately

We also tried several other methods that did not have a clear positive effect and were thus omitted from the final system. For example, we attempted to improve alignment accuracy using the discriminative alignment method proposed by [16] training on the 300 hand-aligned sentences.[4] However, while this provided small gains in alignment accuracy on a held-out set, the gains were likely not enough, and MT results were inconclusive. We also attempted to use the reordering method of [17] as implemented in lader,[5] again trained on the same 300 hand-aligned sentences, but increases in reordering accuracy on a held-out set were minimal. We believe that both of these techniques are promising, but require a larger set of hand-aligned data to provide gains large enough to appear in MT results.

[4] `http://user.phil-fak.uni-duesseldorf.de/~tosch/downloads.html`

[5] `http://phontron.com/lader`

4. Experiments

4.1. Setup

4.1.1. System overview

We used three individual SMT systems for each language pairs: forest-to-string (F2S), hierarchical phrase-based (Hiero), and phrase-based with pre-ordering (Preorder). In some of our comparisons we also use simple phrase-based translation without preordering (PBMT). F2S was implemented with Travatar [18] and Preorder, PBMT, and Hiero were implemented using Moses [19].

For the Moses models, we generally used the default settings, but with Good-Turing phrase table smoothing. For F2S translation we used Egret[6] as a parser, and created forests using dynamic pruning including all edges that occurred in the 100-best hypotheses. We trained the parsing model using the Berkeley parser over the Wall Street Journal section of the Penn Treebank[7] for English, and TIGER corpus [20] for German. For model training, the default settings for Travatar were used, with the exception of changing the number of composed rules to 6 and using Kneser-Ney rule table smoothing.

All systems were evaluated using the standard BLEU score [21] and also RIBES [22], a metric designed specifically to show whether reordering is being performed properly. All systems were optimized towards BLEU score. We measure statistical significance between results with bootstrap resampling with $p > 0.05$. Bold numbers in each table indicate the best system, and all systems that do not show a statistically significant difference from the best system [23].

All words were lowercased prior to translation, and finally recased by a SMT-based recaser as implemented in Moses.

4.1.2. Translation models

We trained the translation models using WIT^3 training data (138,499 sentences) and 1,000,000 sentences selected over other bitexts (Europarl, News Commentary, and Common Crawl) by the method described in 3.1.

4.1.3. Language models

We used two types of word n-gram language models of German and English: interpolated 6-gram and Google 5-gram.

The interpolated 6-gram LMs were from linear interpolation of several 6-gram LMs on different data sources (WIT^3, Europarl, News Commentary, Common Crawl, Common News, and MultiUN). The interpolation weights were optimized for test set perplexities on the development set, using `interpolate-lm.perl` in Moses. Individual 6-gram LMs were trained by SRILM with modified Kneser-Ney smoothing.

[6] `https://github.com/neubig/egret/`
[7] `http://www.cis.upenn.edu/~treebank/`

System	tst2011	tst2012	tst2013
Combination	26.04	22.86	24.60
F2S	26.27	22.59	24.34
Hiero	24.55	20.66	22.80
Preorder	25.30	21.84	24.08

Table 1: Official BLEU results for English-to-German (case-sensitive).

The Google 5-gram LMs were from Google Web 1T N-grams. We limited vocabulary words to those with 8,192 or more in unigram counts and all words were mapped to lowercase. Then we trained 5-gram LMs with Witten-Bell smoothing.

4.1.4. Recaser models

The Moses-based recaser model for both English and German were trained by `train-recaser.perl` using monolingual resources (WIT^3, Europarl, News Commentary, Common Crawl, Common News, and MultiUN).

4.2. Full System Results

Our full system was the combination of F2S, Hiero, and Preorder. Tables 1 and 2 show the evaluation results for the official test sets in German-to-English and English-to-German, respectively. In German-to-English, each individual system showed similar performance in BLEU and the system combination achieved much higher BLEU score, 2.8 points higher than Preorder. In English-to-German, F2S showed the best performance among the three individual systems and the system combination was not so effective as in German-to-English.

The contributions of individual systems can be measured by the number of each system's output chosen by the system combination, as shown in Table 3. These results suggest:

- When one system is much better than the others, our system combination highly relies on the best system and has a little room for improvement. (English-to-German)

- When the individual systems are different each other, the voting-like effect of our system combination improves the overall performance even if individual performances are similar. (German-to-English)

These findings are similar to our system combination results in English-Japanese translation [24].

With respect to recasing, slight BLEU drops were found between case-sensitive and case-insensitive evaluation as shown in Table 4. There was a larger drop in English-German than German-English, due to the large number of required recasing for German nouns.

System	tst2013
Combination	25.83
F2S	23.03
Hiero	22.76
Preorder	23.04

Table 2: Official BLEU results for German-to-English (case-sensitive, without disfluency).

Task	F2S	Hiero	Preorder	ALL
English-German	868	0	125	993
German-English	304	142	916	1,362

Table 3: Number of each system's outputs chosen by system combination for tst2013.

	En-De	De-En
case-sensitive	24.60	25.83
case-insensitive	25.79	26.45

Table 4: Official BLEU results by Combination systems on tst2013 set with case-sensitive and case-insensitive evaluation (without disfluency).

4.3. Effect of Data Selection

Experimental results on adaptation training data selection is shown in Table 5. By adding 1 million (1M) general-domain sentences, we improve a baseline de-en PBMT system (which is only trained from in-domain TED data) from 27.26 to 28.09 BLEU. We improve from 21.53 to 22.11 BLEU in the en-de PBMT system. This 1M general-domain data is combined with the in-domain TED bitext in subsequent system building, which required sufficiently fewer computational resources than using the entire general-domain data (especially for the F2S system).

Interestingly, we have found the improvements in Table 5 are not as large as that reported in [7] despite the similar task setup. The results are not directly comparable due to different dev/test splits and random initializations. Nevertheless, it has come to our attention that the random sampling of general-domain data for $GEN_E(e)$ and $GEN_F(f)$ in Eq. 1 appears to cause large differences in the subsequent RNNLMs. This is because the RNNLMs are highly optimized on perplexity. We suspect that using only $IN_E(e)$ and $IN_F(f)$ as the sentence selection criteria (or using the simpler n-grams for $GEN_E(e)$ and $GEN_F(f)$ values) may give more stable results, though we have not tried comprehensive experiments to validate this.

4.4. Translation Method Comparison

In this section, we provide a brief comparison of the three translation methods mentioned in Section 2 on tst2010 data. For all systems we used the TED data and 1M selected sentences for training, and used the language model described in the previous section. None of the results include RNNLM, and are somewhat preliminary, so they do not match our final submission exactly.

	Number of Selected General-domain Sentences					
	0	100k	500k	1M	2M	all
de-en	27.26	27.51	27.55	28.09	27.43	27.44
en-de	21.53	21.58	21.73	22.11	21.92	22.09

Table 5: BLEU results for adaptation training data selection. These are tst2010 results using a preliminary PBMT system, so they are not directly comparable to other results in this paper.

	en-de		de-en	
	BLEU	RIBES	BLEU	RIBES
PBMT	23.11	80.56	**30.51**	**84.68**
Hiero	23.33	**81.17**	**30.54**	**84.51**
F2S	**24.30**	**81.09**	**30.37**	83.44

Table 6: A comparison between different translation methods with exactly matched training conditions.

	Baseline	+Splitting
PBMT	30.36	**30.51**
Hiero	30.22	**30.54**
F2S	29.82	**30.36**

Table 7: BLEU results for compound splitting.

The results are shown in Table 6. From these results, we can see that given exactly the same data, alignments, and language model, F2S achieved the highest accuracy on English-German, and PBMT and Hiero achieved higher accuracy on German-English. For English-German, we noticed that the F2S system did a significantly better job of accurately generating verbs at the end of the German sentence, demonstrating its superior capability for reordering. For German-English, on the other hand, F2S achieved a somewhat counter-intuitive low score on the reordering-based measure RIBES. Upon an analysis of the results, we found that the F2S system was largely getting the reordering right, but occasionally making big changes in reordering large clauses that were not reflected in the German reference. It is likely that if we optimized towards RIBES, or a combination of BLEU and RIBES [25] we might get better results.

4.5. Effect of Compound Splitting

Next, we examine the effect of compound splitting for German-English translation. From the results in Table 7, we can see that compound splitting provides a gain for all systems, and particularly so for F2S translation.

	en-de		de-en	
	n-gram	+RNNLM	n-gram	+RNNLM
PBMT	23.11	**23.81**	30.51	**31.03**
Hiero	23.33	**24.31**	30.54	**31.80**
F2S	24.30	**25.02**	30.48	**30.85**

Table 8: BLEU results for RNNLM rescoring.

4.6. Effect of RNNLM

Next, we examine the effect of adding RNNLM to the translation accuracy. From the results in Table 8, we can see that the RNNLM provided significant gains in all cases, ranging from 0.4-1.3 BLEU points. Examining the results manually, we it was difficult to identify one clear reason for the improvements in the scores, but we did see some subjective improvements in agreement between prepositions in coordinate structures, and less collapse of syntactic structure around unknown words.

5. Conclusion

We used various SMT technologies for this year's evaluation campaign. Most of them had positive effects on the final translation performance. The forest-to-string SMT had the largest contribution in English-to-German, and the GMBR system combination largely increased the performance in German-to-English.

6. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *Proc. IWSLT 2013*, 2013.

[2] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT*, Edmonton, Canada, 2003, pp. 48–54.

[3] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007.

[4] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. ACL*, 2006.

[5] H. Mi and L. Huang, "Forest-based translation rule extraction," in *Proc. EMNLP*, 2008, pp. 206–214.

[6] J. Graehl and K. Knight, "Training tree transducers," in *Proc. HLT*, 2004, pp. 105–112.

[7] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proc. ACL*, 2013.

[8] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proc. EMNLP*, 2011, pp. 355–362.

[9] M. Collins, P. Koehn, and I. Kucerova, "Clause restructuring for statistical machine translation," in *Proc. ACL*, 2005.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. 11th InterSpeech*, 2010, pp. 1045–1048.

[11] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proc. EACL*, 2003.

[12] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Generalized minimum bayes risk system combination," in *Proc. IJCNLP*, 2011, pp. 1356–1360.

[13] K. Sudoh, K. Duh, H. Tsukada, M. Nagata, X. Wu, T. Matsuzaki, and J. Tsujii, "NTT-UT statistical machine translation in NTCIR-9 PatentMT," in *Proc. NTCIR*, 2011.

[14] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 217–226.

[15] C. He, C. Wang, Y.-X. Zhong, and R.-F. Li, "A survey on learning to rank," in *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 3. IEEE, 2008, pp. 1734–1739.

[16] J. Riesa and D. Marcu, "Hierarchical search for word alignment," in *Proc. ACL*, 2010, pp. 157–166.

[17] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering," in *Proc. EMNLP*, Korea, July 2012, pp. 843–853.

[18] G. Neubig, "Travatar: A forest-to-string machine translation engine based on tree transducers," in *Proc. ACL Demo Track*, Sofia, Bulgaria, August 2013.

[19] P. 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, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL*, Prague, Czech Republic, 2007, pp. 177–180.

[20] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit, "Tiger: Linguistic interpretation of a german corpus," *Research on Language and Computation*, vol. 2, no. 4, pp. 597–620, 2004.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, Philadelphia, USA, 2002, pp. 311–318.

[22] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs," in *Proc. EMNLP*, 2010, pp. 944–952.

[23] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. EMNLP*, 2004.

[24] K. Sudoh, H. Tsukada, M. Nagata, S. Hoshino, and Y. Miyao, "NTT-NII statistical machine translation in NTCIR-10 PatentMT," in *Proc. NTCIR*, 2013.

[25] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Learning to translate with multiple objectives," in *Proc. ACL*, 2012.