The NAIST Machine Translation System for IWSLT2012

Graham Neubig, Kevin Duh, Masaya Ogushi, Takamoto Kano
Tetsuo Kiso, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science
Nara Institute of Science and Technology, Japan

Abstract

This paper describes the NAIST statistical machine translation system for the IWSLT2012 Evaluation Campaign. We participated in all TED Talk tasks, for a total of 11 language-pairs. For all tasks, we use the Moses phrase-based decoder and its experiment management system as a common base for building translation systems. The focus of our work is on performing a comprehensive comparison of a multitude of existing techniques for the TED task, exploring issues such as out-of-domain data filtering, minimum Bayes risk decoding, MERT vs. PRO tuning, word alignment combination, and morphology.

1. Introduction

This paper describes the NAIST participation in the IWSLT 2012 evaluation campaign [1]. We participated in all 11 TED tasks, dividing our efforts in half between the official English-French track and the 10 other unofficial Foreign-English tracks. For all tracks we used the Moses decoder [2] and its experiment management system to run a large number of experiments with different settings over many language pairs.

For the English-French system we experimented with a number of techniques, settling on a combination that provided significant accuracy improvements without introducing unnecessary complexity into the system. In the end, we chose a four-pronged approach consisting of using the web data with filtering to remove noisy sentences, phrase table smoothing, language model interpolation, and minimum Bayes risk decoding. This led to a score of 31.81 BLEU on the tst2010 data set, a significant increase over 29.75 BLEU of a comparable system without these improvements. In Section 2 we describe each of the methods in more detail and examine their contribution to the accuracy of the system. For reference purposes, in Section 3, we also present additional experiments that gave negative results, which were not included in our official submission.

For the 10 translation tasks into English, we focused on techniques that could be used widely across all languages. In particular, we experimented with unsupervised approaches to handling source-side morphology, minimum Bayes risk decoding, and large language models. In the end, most of our systems used a combination of unsupervised morphology processing and large language models, which resulted in an average gain of 1.18 BLEU points over all languages. Section 4 describes these results in further detail.

Decoding	dev2010	tst2010
Baseline	26.02	29.75
NAIST Submission	27.05	**31.81**

Table 1: The scores for systems with and without the proposed improvements.

2. English-French System

The NAIST English-French translation system for IWSLT 2012 was based on phrase-based statistical machine translation [3] using the Moses decoder [2] and its corresponding training regimen. Overall, we made four enhancements over the standard Moses setup to improve the translation accuracy:

Large-scale Data with Filtering: In order to use the large, but noisy parallel training data in the English-French Giga Corpus, we implemented a technique to filter out noisy translated text.

Phrase Table Smoothing: We performed phrase table smoothing to improve the probability estimates of low-frequency phrases.

Language Model Interpolation: In order to adapt to the domain of the task, we interpolated language models trained using text from several domains.

Minimum Bayes-Risk Decoding: We used lattice-based minimum Bayes risk decoding to select hypotheses that are supported by other hypotheses in the n-best list, and calibrated the probability distribution to further improve performance.

We demonstrate our results (in BLEU score) before and after these techniques are added in Table 1. It can be seen that the combination of these 4 improvements leads to a 2.06 point gain in BLEU score on tst2010 over the baseline system. We will explain each of the techniques in detail as follows.

Corpus	English	French
TED	2.36M	2.47M
News Commentary (NC)	2.99M	3.45M
EuroParl (EP)	50.3M	52.5M
United Nations (UN)	302M	338M
WMT2012 Giga	575M	672M
Giga (+Filtering)	485M	565M

Table 2: The number of words in each corpus.

2.1. Data

The first step of building our system was preparing the data. Table 2 shows the size and genre of each of the corpora available for the task. From these corpora, we used TED, NC, EuroParl, UN, and Giga for training the language model, and TED, NC, EuroParl, and filtered Giga (explained below) for training the translation model.[1] Tuning was performed on dev2010, and testing was performed on tst2010.

In particular, the English-French Giga-word corpus is from the web and thus covers a wide variety of diverse topics, making it a strong ally for the construction of a general domain machine translation system. However, as the sentences were automatically extracted, they contain a significant number of errors where the content of the parallel sentences actually do not match, or only match partially. In order to filter out some of this noise, we re-implemented a variant of the sentence filtering method of [4].

The method works by using a clean corpus to train a classifier that can detect mis-aligned sentences. Because the clean corpus only contains correctly aligned sentences, we create pseudo-negative examples by traversing the corpus and randomly swapping two consecutive sentences with some set probability. These swapped sentences are labeled as "negative," and the remainder of the unswapped samples are labeled as positive.

In this application, the feature set chosen for the classifier must satisfy two desiderata. First, as with all machine learning applications, the features must be sufficient to discriminate between the classes that we are interested in: properly or improperly aligned sentences. Second, as our training data (a clean corpus) and testing data (a noisy corpus) will necessarily be drawn from different domains, we would like to use a small, highly generalizable feature set that will work on both domains. In order to achieve both of these objectives, we take hints from [4] and [5] to define the following features, where f_1^J and e_1^I are the source and target sentences, and J and I are their respective lengths:

Length Ratio features capture the fact that properly aligned sentences should be approximately the same length. Two continuous features $\max(J, I)/\min(J, I)$, J/I, and three indicator features $J > I$, $I > J$, $I = J$.

Giga Data	dev2010	tst2010
None	26.61	31.52
Unfiltered	27.03	**31.90**
Filtered	27.05	31.81

Table 3: Accuracy given various styles of using the Giga data.

Model One Probability features capture the fact that an unsupervised alignment model (in this case, the efficiently calculable IBM Model One [6]) should assign higher probability to well-aligned sentences. In this category, we use two continuous features $\log P_{M1}(e_1^I|f_1^J)$ and $\log P_{M1}(f_1^J|e_1^I)$.

Alignment features use Viterbi word alignments and capture certain patterns that should occur in properly aligned sentences. Word alignments are calculated using IBM Model One, and symmetrized using the "intersection" criterion [7]. If the number of aligned words is K, our features include aligned word ratio $K/\min(I, J)$, total number of aligned words K, number of alignments that are monotonic, monotonic alignment ratio, and the average length of gaps between words (similar to "distortion" used in phrase-based MT [3]).

Same Word features count the number of times that a word of length n is exactly equal to a word in the opposite sentence. This is useful for noticing when proper names, numbers, or words with a shared linguistic origin occur in both sentences. In our system we use separate features for $n = 1$, $n = 2$, $n = 3$, and $n \geq 4$.

To train the non-parallel sentence identifier, we use data from the TED, NC, and EuroParl corpora swapping sentences with a probability of 0.3 to create pseudo-negative examples. We use this as training data for a support vector machine (SVM) classifier, which we train using LIBLINEAR [8]. In order to get an estimate of the accuracy of sentence filtering, we perform 8-fold cross validation on the training data, and achieve a classification accuracy of 98.0%.[2]

Next, we run the trained classifier on the entirety of the Giga corpus and remove the examples labeled as non-parallel. As a result of filtering with the classifier, a total of 485M English and 565M French words remained, a total of 84.3% of the original corpus.

Finally, using no Giga data, the unfiltered Giga data, and the filtered Giga data (in addition to all other data sets), we measured the final accuracy of the translation system. The

[1] We also attempted to use the UN corpus for training the translation model, but found that it provided no gain, likely because of the specialized writing style of UN documents.

[2] Of course, as we are using pseudo-negative examples in the Europarl corpus instead of real negative examples from the Giga corpus, these accuracy features are only approximate.

Smoothing	dev2010	tst2010
None	26.75	31.19
Good-Turing	27.05	**31.81**

Table 4: BLEU results using translation model smoothing.

LM	dev2010	tst2010
TED Only	24.80	29.44
Without Interp.	26.30	31.15
With Interp.	27.05	**31.81**

Table 5: Results training the language model on only TED data, and when other data is used without and with language model interpolation.

results are shown in Table 3. As a result, we can see that using the data from the Giga corpus has a positive effect on the results, but filtering does not have a clear significant effect on the results.

2.2. Phrase Table Smoothing

We also performed experiments that used smoothing of the statistics used in calculating translation model probabilities [9]. The motivation behind this method is that the statistics used to train the phrase table are generally sparse, and tend to over-estimate the probabilities of rare events. In the submitted system we used Good-Turing smoothing for the phrase table probabilities.

Results comparing a system with smoothing and without smoothing can be found in Figure 4. It can be seen that Good-Turing smoothing of the phrase table improves results by a significant amount.

2.3. Language Model Interpolation

One of the characteristics of the IWSLT TED task is that, as shown in Table 2, we have several heterogeneous corpora. In addition, the in-domain TED data is relatively small, so it can be expected that we will benefit from using data outside of the TED domain. In order to effectively utilize out-of-domain data in language modeling, we build one language model for each domain and interpolate the language models to minimize perplexity on the TED dev2010 set using the method described by [10] and implemented in the SRILM toolkit [11].

To measure the effectiveness of this technique, we also measure the accuracy without any data other than TED, and when the data from all domains was simply concatenated together for LM learning. The results can be found in Table 5. We can see that adding the larger non-TED data to the language model is essential, and using linear interpolation to adjust the language model weights can also provide large further gains.

2.4. Minimum Bayes Risk Decoding

Finally, we experimented with improved decoding strategies for translation, particularly using minimum Bayes risk decoding (MBR, [12]). In normal translation, the decoder attempts to simply find the answer with the highest probability among the translation candidates

$$\hat{E} = \text{argmax}_E P(E|F) \tag{1}$$

in a process called Viterbi decoding. As an alternative to this, MBR attempts to find the hypothesis that minimizes risk

$$\hat{E} = \text{argmin}_E \sum_{E' \in \mathcal{E}} P(E'|F) L(E', E) \tag{2}$$

considering the posterior probability $P(E'|F)$ of hypotheses E' in the space of all possible hypotheses $\mathcal{E}$, as well as a loss $L(E', E)$ which determines how bad a translation E is if the true translation is E'. In this work (as with most others on MBR in MT) we use one minus sentence-wise BLEU+1 score [13] as our loss function

$$L(E', E) = 1 - \text{BLEU+1}(E', E). \tag{3}$$

In initial research on MBR, the space of possible hypotheses $\mathcal{E}$ was defined as the n-best list output by the decoder. This was further expanded by [14], who defined MBR over lattices. We tested both of these approaches (as implemented in the Moses decoder).

Finally, one fine point about MBR is that it requires a good estimate of the probability $P(E'|F)$ of hypotheses. In the discriminative training framework of [15], which is used in most modern SMT systems, scores of machine translation hypotheses are generally defined as a log-linear combination of feature functions such as language model or translation model probabilities

$$P(E'|F) = \frac{1}{Z} e^{\sum_i w_i \phi_i(E',F)} \tag{4}$$

where ϕ_i indicates feature functions such as the language model, translation model, and reordering model log probabilities, w_i is the weight measuring the relative importance of this feature, and Z is a partition function that ensures that the probabilities add to 1.

Choosing the weights w_i for each feature such that the answer with highest probability

$$\hat{E} = \text{argmax}_E P(E|F) \tag{5}$$

is the best possible translation is a process called "tuning," and essential to modern SMT systems. However, in most tuning methods, including the standard minimum error rate training [16] that was used in the proposed system, while the relative weight of each feature w_i is adjusted, the overall sum of the weights $\sum_i w_i$ is generally set fixed at 1. While this is not a problem when finding the highest probability hypothesis in 5, it will affect the probability estimates $P(E'|F)$, with

Decoding	dev2010	tst2010
Viterbi	27.59	31.01
MBR ($\lambda = 1$)	27.29	31.24
Lattice MBR ($\lambda = 1$)	26.70	31.25
Lattice MBR ($\lambda = 5$)	27.05	**31.81**

Table 6: BLEU Results using Minimum Bayes Risk decoding.

larger s assigning a larger probability to the most probable hypothesis, and a smaller s spreading the probability mass more evenly across all hypotheses.

In order to improve the calibration of our probability estimates, and thus improve the performance of MBR, we introduce an addition scaling factor λ into the calculation of our probability

$$P(E'|F) = \frac{1}{Z} e^{\lambda \sum_i w_i \phi_i(E',F)}. \tag{6}$$

Using this lambda, we tried every value in 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, and 10.0, and finally chose $\lambda = 5.0$, which gave the best performance on tst2010.

The final results of our system with Viterbi decoding (no MBR), regular MBR over n-best lists, and lattice MBR with the scaling factors of 1 and 5, are shown in Table 6. It can be seen that both MBR and lattice-based MBR give small improvements over the baseline without tuning λ, while tuning λ gives a large improvement.[3] The reason why MBR reduces the accuracy on dev2010 is because dev2010 was used in tuning the parameters during MERT, so the one-best answers tend to be better on average than they would be on a held-out test set.

3. Additional Results on English-French

This section presents additional results obtained on the English-French track. The results here, for the most part, did not obtain worthwhile BLEU improvements in preliminary experiments, so we did not include them in the official system as described in Section 2. Although the systems reported in this section use the same dev and test set as that of Section 2, the training conditions and system configurations have slight differences, so the results should not be directly compared. We include these (negative) results for reference purposes, in order to aid understanding of the English-French TED task.

3.1. Exploiting Out-of-domain Data

We experimented with the simplest approach to exploiting out-of-domain bitext in translation models: data concatenation. This can be seen as adaptation at the earliest stage of the translation pipeline, and has achieved competitive results on TED En-Fr [17]. Three conditions were tried: (1) TED-only data, (2) TED + News (NC), (3) TED + NC + EuroParl (EP). Results are shown in Table 7.

First, we observe that adding data gives slight improvements (29.32 to 29.57). To analyze the potential for improvement, we also measured BLEU using "CheatLM" decoding [18]. "CheatLM" is an analysis technique for TM adaptation where the language model is trained on the reference; this gives a optimistic estimate on what can be achieved by the translation model, if other components are tuned almost perfectly. Here we see that TED+NC+EP (59.93 BLEU) can achieve large improvements over TEDonly (55.10 BLEU), indicating the potential value of out-of-domain bitext. However, note that the corresponding OOV rate reduction is relatively small (1.2% to 0.52%). We hypothesize that out-of-domain probably is not helping because of improved word coverage, but rather because of improved word alignment estimation. In any case, the improvements are slight so we do not attempt to draw any further conclusions.

Data	standard	CheatLM	force	OOV
TEDonly	29.32	55.10	16%	1.2%
TED+NC	29.43	58.64	17%	0.85%
TED+NC+EP	**29.57**	**59.93**	21%	0.52%

Table 7: Translation Model Adaption by simple out-of-domain data concatenation. The "standard" and "CheatLM" columns show the BLEU scores on tst2012, using standard Moses decoding and "CheatLM" decoding. The column "force" shows the percentage of tst2010 sentences that can be translated into the reference using forced decoding. OOV indicates the token out-of-vocabulary rate.

3.2. Word Alignment & Phrase Table Combination

We investigated different alignment tools and ways to combine them, as shown in Table 8. Observations are as follows:

- GIZA++ and BerkeleyAligner achieve similar BLEU on this task.
- Concatenating GIZA++ and BerkeleyAligner word alignment results, prior to phrase extraction, achieves a small boost (29.57 to 29.89 BLEU).
- We also experimented with pilaign [19], a Bayesian phrasal alignment toolkit. This tool directly extracts phrases without resorting to the preliminary step of word alignments, and achieves extremely compact phrase table sizes (0.8M entries) without significantly sacrificing BLEU (29.24).
- Combining the GIZA++ and pialign phrase tables by Moses' multiple decoding paths feature did not improve results. Overall, we did not find much differ-

[3] It should be noted that due to constraints in the available data for these MBR experiments we are both tuning on testing on tst2010, but the tuning of λ also demonstrated gains in accuracy on the official blind test on tst2011 and tst2012 (37.33→37.90 and 38.92→39.47 respectively).

ence among these various approaches so we used the standard GIZA++ tool chain in the official submission.

Tool	BLEU	TableSize
1: GIZA++	29.57	109
2: BerkeleyAligner	29.39	170
3: pialign	29.24	0.8
1+2: ConcatAlign (GIZA,Berkeley)	**29.89**	200
1+3: TwoTable (GIZA,pialign)	29.56	201

Table 8: BLEU scores on tst2010 of various combinations of alignment and phrase training tools. TableSize shows the phrase-table size of corresponding method (in millions of entries). GIZA++ and BerkeleyAligner are trained the the TED+NC+EP bitext; pialign is trained only on TED, due to time constraints in our preliminary experiments.

3.3. Lexical Reordering Models

Several reordering models available in the Moses decoder were tried. In general, we found the full "msd-bidir-fe" option to perform best, despite the small number of word order differences between English and French. Results are shown in Table 9.

Reordering model	BLEU
msd-bidir-fe	**29.57**
msd-bidir-f	29.43
monotonicity-bidir-fe	29.29
msd-backward-fe	29.22
distance	28.99
msd-bidir-fe-collapse	28.86

Table 9: Comparison of Reordering models on tst2010.

3.4. MERT vs. PRO tuning

We compared two tuning methods: MERT and PRO [20]. We used the implementations distributed with Moses. For both MERT and PRO, we set the size of k-best list to $k = 100$, used 14 standard features, and removed duplicates in k-best lists when merging previously generated k-best lists. We ran MERT in multi-threaded setting until convergence. Since the number of random restarts in MERT greatly affects on the translation accuracy [21], we tried various number of random restarts for 1, 10, 20, and 50.[4] For PRO, we used MegaM[5] as a binary classifier with the default setting. We ran PRO for 25 iterations. We tried two kinds of PRO: [20] interpolated the weights with previously learned weights to improve the stability (henceforth "PRO-interpolated")[6], and the version that do not use such a interpolation (henceforth "PRO-basic").

			Time (m)	
# of random restarts	Iteration	Dev BLEU	Wall	CPU
1	11	28.18	0.59	0.82
10	11	28.17	2.21	17.22
20	12	28.29	4.91	57.88
50	12	28.31	9.72	171.91

Table 10: The effect of the number of random restarts in MERT on BLEU score and multi-threaded time. "Iteration" denotes the number of iterations which MERT needs to be converged. "Time" denote the average time of weight optimization for each iteration, averaged over all iterations.

Method	Dev BLEU
MERT	**28.29**
PRO-basic	26.99
PRO-interpolated	27.11

Table 11: Comparison with MERT and PRO. For MERT, the number of random restarts was set to 20.

We first investigate the effect of the number of random restarts in MERT on BLEU score and run-time for each iteration. Table 10 shows the result. As the number of random restarts increases, BLEU score improves. However, the run-time increases as well. We used 20 random restarts to compare to PRO.

Table 11 shows the results of MERT and PRO. As can be seen in Figure 11, MERT exceeds PRO-basic by 1.3 points and PRO-interpolated by 1.18 points. As a result, we used MERT for tuning in Sections 2 and 4.

4. Systems for Translation into English

We participated in the translation of all 10 additional language-pairs of the TED Talk track. The source languages are Arabic (ar), German (de), Dutch (nl), Polish (pl), Brazilian-Portuguese (pt), Romanian (ro), Russian (ru), Slovak (sk), Turkish (tr), and Chinese (zh). The target language for all tasks is English (en).

Since all tasks translate into the same language, we are able to share the language model as well as many of the configurations for the Experimental Management System (EMS). This setup provides an invaluable chance to compare the same techniques across structurally-different languages, and is the focus of our work. Rather than optimizing for specific languages, we concentrate on building common systems under the same EMS framework and on comparing the performance of existing techniques cross-lingually.

It is interesting to note that the 10 language-pairs cover a diverse range of linguistic phenomenon. In terms of historical relationships, the Italic family (pt,ro) and Germanic family (de, nl) are expected to be closer to the target language of English. The Slavic family (pl,ru,sk), Arabic, and Turkish

[4] Currently, Moses's default setting is 20.

[5] `http://www.cs.utah.edu/~hal/megam/`

[6] We set the same interpolation coefficient value of 0.1 as [20] noted.

languages exhibit rich morphology (fusional, non-catenative, or agglutinative). Additionally, the Germanic family may show word order differences (V2 and SOV) and Chinese requires word segmentation.

4.1. Experiments

Table 12 summarizes all the results (BLEU scores) for translation into English. In all language pairs, the baseline consists of a standard phrase-based Moses system (GIZA++ alignment, grow-diag-final-and heuristic, lexical ordering, 4-gram language model) trained on the TED Talks portion of the training data. MERT tuning is performed on the "dev2010" portion of the data and Table 12 shows test results on "tst2010."[7] While it is not possible to directly compare BLEU across languages, we do observe that the Italic and Germanic languages fare better on this TED task (> 25 BLEU), while Chinese, Turkish, and the Slavic languages perform poorly at $10 - 17$ BLEU.

We then proceeded to improve on these baseline results. First, adding additional out-of-domain data (nc=News Commentary, ep=Europarl, un=UN Multitext) to the language model increased results uniformly for all language pairs (line (b) of Table 12). We used an interpolated language model, trained in the same fashion as in our English-French system.

Next, we tried two strategies for handling rich morphology in the input. The "CompoundSplit" program in the Moses package was developed for languages with extensive noun compounding, e.g. German, and breaks apart words if sub-parts are seen in the training data over a certain frequency [22]. The alternate "Morfessor" program [23] is an unsupervised morphological analyzer based on the Minimum Description Length principle – it tries to find the the smallest set of morphemes that parsimoniously cover the training set. Morfessor is expected to segment more aggressively than CompoundSplit, especially because it can find both bound and free morphemes. However, we empirically found that Morfessor segments too aggressively for unknown words (i.e. each character becomes a morpheme), so we do not segment OOV words in dev/test.[8] The results in line (c) of Table 12 shows that German benefit most from CompoundSplit, while Arabic, Russian, and Turkish benefit from Morfessor. The remaining languages perform approximately equal or slightly better with these morphology enhancements, so in further experiments we keep the morphology pre-processing (de & ro uses CompoundSplit; others use Morfessor).

In line (d) of Table 12, we further added the Giga corpus to the interpolated language model. For some languages, this gave a large improvement (ar, de, pl, sk), while for other languages the results remain similar. Some of these results represent our official submission. In line (e), adding Lattice MBR decoding uniformly degraded results, so we chose not to include it. This is in contrast with our English-French results. We suspect that in this case uniformity of the training data and lack of diversity in the n-best list may have damaged MBR; the resulting translations appear similar in structure, but many have extraneous articles and determiners, which hurts BLEU. It should also be noted that unlike English-French, we did not calibrate the probability distribution by adjusting λ, which might also had a significant effect on the results. Finally in line (f), we added additional out-of-domain bitext for Translation Model training. This only helped slightly for pl and tr, while degrading other language pairs: we conclude that more advanced TM adaptation methods is necessary, and simply concatenating the bitext does not help.

Finally, we note that our submitted systems for each language achieve a 0.7-2.5 BLEU improvement over the respective baselines. We also achieve slight improvements in METEOR, despite not tuning for it. While the feature that helped most depends on language, we observe that morphological pre-processing and larger language models are generally worthwhile efforts.

5. Conclusion

This paper described our experiments with a number of existing machine translation techniques for the IWSLT 2012 TED task. Some of these techniques, such as minimum Bayes risk decoding with calibrated probabilities, language model interpolation, unsupervised morphology processing, translation model smoothing, and the use of large data proved to be effective. We also found that a number of techniques, including tuning using PRO, alignment combination, and data filtering had less of a positive effect.

6. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, HK, December 2012.

[2] 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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007, pp. 177–180.

[3] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, 2003, pp. 48–54.

[4] D. S. Munteanu and D. Marcu, "Improving machine translation performance by exploiting non-parallel corpora," *Computational Linguistics*, vol. 31, no. 4, pp. 477–504, 2005.

[7] For Slovak, which lacked an official dev/test split, we split the development data, with the first half for tuning and the second half for testing. All source languages, except for Slovak, have comparable amounts of in-domain data (130k-145k sentence pairs).

[8] In other words, we keep OOV words as is and propagate it to the output. This implies that we lose the opportunity to translate OOV words whose component morphemes are seen in the training data. However, we think this conservative option is safer in the presence of potential over-segmentation.

SYSTEM	ar	de	nl	pl	pt	ro	ru	sk	tr
(a): baseline	21.6	26.8	30.6	15.5	35.6	28.7	16.8	16.8	12.5
(b): (a)+LM:nc,ep,un	21.9	26.9	31.4	15.6	36.1	29.2	17.3	17.7	12.6
(c): (b)+morphology									
compoundsplit	22.5	27.4	31.2	15.6	36.2	29.1	17.0	17.7	12.9
morfessor	23.4	26.8	31.6	15.6	36.3	28.8	17.6	17.7	13.6
(d): (c)+LM:giga	**24.1**	**28.0**	**31.4**	16.2	**36.2**	**29.4**	**17.5**	**18.4**	13.8
(e): (d)+lattice MBR	23.4	27.1	30.7	15.4	34.7	27.6	16.4	17.8	13.7
(f): (d)+TM (outdomain)	21.7	26.2	29.5	**16.4**	35.3	29.3	16.5	-	**13.9**
Δbleu: (d) or (f) - (a)	2.5	1.2	0.8	0.9	0.7	0.7	0.8	1.6	1.4
Δmeteor : (d) or (f) - (a)	1.7	0.7	0.2	0.7	0.2	0.3	0.8	0.6	1.5

Table 12: BLEU Results for Translations into English. Roughly, each row builds on top of the previous row. Boldface indicates official submission. For zh-en (not shown in table as the segmentation methods are different from other language pairs), the BLEU results are: 10.8 for character-based translation and 11.6 for word-based translation (Stanford word segmenter, PKU standard), using +LM:nc,ep,un but not +LM:giga nor +TM(outdomain), which degraded results.

[5] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French translation systems for IWSLT 2011," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[6] P. F. Brown, V. J. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, pp. 263–312, 1993.

[7] F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation," *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20–28, 1999.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, 2008.

[9] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 53–61.

[10] F. Jelinek and R. L. Mercer, "Interpolated estimation of markov source parameters from sparse data," pp. 381–397, 1980.

[11] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Speech and Language Processing (ICSLP)*, 2002.

[12] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (NAACL) Meeting (HLT/NAACL)*, 2004.

[13] C.-Y. Lin and F. J. Och, "Orange: a method for evaluating automatic evaluation metrics for machine translation," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, 2004, pp. 501–507.

[14] R. Tromble, S. Kumar, F. Och, and W. Macherey, "Lattice Minimum Bayes-Risk decoding for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 620–629.

[15] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[16] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003.

[17] K. Duh, K. Sudoh, and H. Tsukada, "Analysis of translation model adaptation for statistical machine translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) - Technical Papers Track*, 2010.

[18] S. Matsoukas, personal communication, 2010.

[19] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine translation without words through substring alignment," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Korea, July 2012, pp. 165–174.

[20] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[21] R. C. Moore and C. Quirk, "Random restarts in minimum error rate training for statistical machine translation," in *Proceedings of the 22th International Conference on Computational Linguistics (COLING)*, 2008, pp. 585–592.

[22] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of the 10th European Chapter of the Association for Computational Linguistics (EACL)*, 2003.

[23] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, 2002, pp. 21–30.