TÜBİTAK-BİLGEM German-English Machine Translation Systems for WMT'13

İlknur Durgar El-Kahlout and Coşkun Mermer
TÜBİTAK-BİLGEM
Gebze 41470, Kocaeli, TURKEY
{ilknur.durgar,coskun.mermer}@tubitak.gov.tr

Abstract

This paper describes TÜBİTAK-BİLGEM statistical machine translation (SMT) systems submitted to the Eighth Workshop on Statistical Machine Translation (WMT) shared translation task for German-English language pair in both directions. We implement phrase-based SMT systems with standard parameters. We present the results of using a big tuning data and the effect of averaging tuning weights of different seeds. Additionally, we performed a linguistically motivated compound splitting in the German-to-English SMT system.

1 Introduction

TÜBİTAK-BİLGEM participated for the first time in the WMT'13 shared translation task for the German-English language pairs in both directions. We implemented a phrase-based SMT system by using the entire available training data. In the German-to-English SMT system, we performed a linguistically motivated compound splitting. We tested different language model (LM) combinations by using the parallel data, monolingual data, and Gigaword v4. In each step, we tuned systems with five different tune seeds and used the average of tuning weights in the final system. We tuned our systems on a big tuning set which is generated from the last years' (2008, 2009, 2010, and 2012) development sets. The rest of the paper describes the details of our systems.

2 German-English

2.1 Baseline

All available data was tokenized, truecased, and the maximum number of tokens were fixed to 70 for the translation model. The Moses open SMT toolkit (Koehn et al., 2007) was used with MGIZA++ (Gao and Vogel, 2008) with the standard alignment heuristic *grow-diag-final* (Och and Ney, 2003) for word alignments. *Good-Turing* smoothing was used for phrase extraction. Systems were tuned on *newstest2012* with MERT (Och, 2003) and tested on *newstest2011*. 4-gram language models (LMs) were trained on the target side of the parallel text and the monolingual data by using SRILM (Stolcke, 2002) toolkit with Kneser-Ney smoothing (Kneser and Ney, 1995) and then binarized by using KenLM toolkit (Heafield, 2011). At each step, systems were tuned with five different seeds with lattice-samples. Minimum Bayes risk decoding (Kumar and Byrne, 2004) and *-drop-unknown* parameters were used during the decoding.

This configuration is common for all of the experiments decribed in this paper unless stated otherwise. Table 1 shows the number of sentences used in system training after the *clean-corpus* process.

Data	Number of sentences
Europarl	1908574
News-Commentary	177712
Commoncrawl	726458

Table 1: Parallel Corpus.

We trained two baseline systems in order to assess the effects of this year's new parallel data, *commoncrawl*. We first trained an SMT system by using only the training data from the previous WMT shared translation tasks that is *europarl* and *news-commentary* (**Baseline1**). As the second baseline, we also included the new parallel data *commoncrawl* only in the translation model (**Baseline2**). Then, we included *commoncrawl* corpus both to the translation model and the language model (**Baseline3**).

Table 2 compares the baseline results. For all

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 109–113,
Sofia, Bulgaria, August 8-9, 2013

experiments throughout the paper, we present the minimum and the maximum BLEU scores obtained after five different tunes. As seen in the table, the addition of the *commoncrawl* corpus resultedin a 1.1 BLEU (Papineni et al., 2002) points improvement (on average) on the test set. Although **Baseline2** is slightly better than **Baseline3**, we used **Baseline3** and kept *commoncrawl* corpus in LMs for further experiments.

System	newstest12	newstest11
Baseline1	20.58|20.74	19.14|19.29
Baseline2	21.37|21.58	20.16|20.46
Baseline3	21.28|21.58	20.22|20.49

Table 2: Baseline Results.

2.2 Bayesian Alignment

In the original IBM models (Brown et al., 1993), word translation probabilities are treated as model parameters and the expectation-maximization (EM) algorithm is used to obtain the maximum-likelihood estimates of the parameters and the resulting distributions on alignments. However, EM provides a point-estimate, not a distribution, for the parameters. The Bayesian alignment on the other hand takes into account all values of the model parameters by treating them as multinomial-distributed random variables with Dirichlet priors and integrating over all possible values. A Bayesian approach to word alignment inference in IBM Models is shown to result in significantly less "garbage collection" and a much more compact alignment dictionary. As a result, the Bayesian word alignment has better translation performances and obtains significant BLEU improvements over EM on various language pairs, data sizes, and experimental settings (Mermer et al., 2013).

We compared the translation performance of word alignments obtained via Bayesian inference to those obtained via EM algorithm. We used a a Gibbs sampler for fully Bayesian inference in HMM alignment model, integrating over all possible parameter values in finding the alignment distribution by using **Baseline3** word alignments for initialization. Table 3 compares the Bayesian alignment to the EM alignment. The results show a slight increase in the development set *newstest12* but a decrease of 0.1 BLEU points on average in the test set *newstest11*.

System	newstest12	newstest11
Baseline3	21.28|21.58	20.22|20.49
Gibbs Sampling	21.36|21.59	19.98|20.40

Table 3: Bayesian Alignment Results.

2.3 Development Data in Training

Development data from the previous years (i.e. *newstest08*, *newstest09*, *newstest10*), though being a small set of corpus (7K sentences), is in-domain data and can positively affect the translation system. In order to make use of this data, we experimented two methods: i) adding the development data in the translation model as described in this section and ii) using it as a big tuning set for tuning the parameters more efficiently as explained in the next section.

Similar to including the *commoncrawl* corpus, we first add the development data both to the training and language models by concatenating it to the biggest corpus *europarl* (**DD(tm+lm)**) and then we removed this corpus from the language models (**DD(tm)**). Results in Table 4 show that including the development data both the tranining and language model increases the performance in development set but decreases the performance in the test set. Including the data only in the translation model shows a very slight improvement in the test set.

System	newstest12	newstest11
Baseline3	21.28|21.58	20.22|20.49
DD(tm+lm)	21.28|21.65	20.00|20.49
DD(tm)	21.23|21.52	20.26|20.49

Table 4: Development Sets Results.

2.4 Tuning with a Big Development Data

The second method of making use of the development data is to concatenate it to the tuning set. As a baseline, we tuned the system with *newstest12* as mentioned in Section 2.1. Then, we concatenated the development data of the previous years with the *newstest12* and built a big tuning set. Finally, we obtained a tuning set of 10K sentences. We excluded the *newstest11* as an internal test set to see the relative improvements of different systems. Table 5 shows the results of using a big tuning set. Tuning the system with a big tuning set resulted in a 0.13 BLEU points improvement.

System	newstest12	newstest11
newstest12	21.28|21.58	20.22|20.49
Big Tune	20.93|21.19	20.32|20.58

Table 5: Tuning Results.

2.5 Effects of Different Language Models

In this set of experiments, we tested the effects of different combinations of parallel and monolingual data as language models. As the baseline, we trained three LMs, one from each parallel corpus as *europarl*, *news-commentary*, and *commoncrawl* and one LM from the monolingual data *news-shuffled* (**Baseline3**). We then trained two LMs, one from the whole parallel data and one from the monolingual data (**2LMs**). Table 6 shows that using whole parallel corpora as one LM performs better than individual corpus LMs and results in 0.1 BLEU points improvement on the baseline. Finally, we trained Gigaword v4 (LDC2009T13) as a third LM (**3LMs**) which gives a 0.16 BLEU points improvement over the **2LMs**.

System	newstest12	newstest11
Baseline3	21.28|21.58	20.22|20.49
2LMs	21.46|21.70	20.28|20.57
3LMs	21.78|21.93	20.54|20.68

Table 6: Language Model Results.

2.6 German Preprocessing

In German, compounding is very common. From the machine translation point of view, compounds increase the vocabulary size with high number of the singletons in the training data and hence decrease the word alignment quality. Moreover, high number of out-of-vocabulary (OOV) words in tuning and test sets results in several German words left as untranslated. A well-known solution to this problem is compound splitting.

Similarly, having different word forms for a source side lemma for the same target lemma causes the lexical redundancy in translation. This redundancy results in unnecessary large phrase translation tables that overload the decoder, as a separate phrase translation entry has to be kept for each word form. For example, German definite determiner could be marked in sixteen different ways according to the possible combinations of genders, case and number, which are fused in six different tokens (e.g., der, das, die, den, dem, des). Except for the plural and genitive cases, all these forms are translated to the same English word "the".

In the German preprocessing, we aimed both normalizing lexical redundancy and splitting German compounds with corpus driven splitting algorithm based on Koehn and Knight (2003). We used the same compound splitting and lexical redundancy normalization methods described in Allauzen et al. (2010) and Durgar El-Kahlout and Yvon (2010) with minor in-house changes. We used only "addition" (e.g., -s, -n, -en, -e, -es) and "truncation" (e.g., -e, -en, -n) affixes for compound splitting. We selected minimum candidate length to 8 and minimum split length to 4. By using the Treetagger (Schmid, 1994) output, we included linguistic information in compound splitting such as not splitting named entities and foreign words (**CS1**). We also experimented adding # as a delimiter for the splitted words except the last word (e.g., Finanzkrisen is splitted as finanz# krisen) (**CS2**).

On top of the compound splitting, we applied the lexical redundancy normalization (**CS+Norm1**). We lemmatized German articles, adjectives (only positive form), for some pronouns and for nouns in order to remove the lexical redundancy (e.g., Bildes as Bild) by using the fine-grained part-of-speech tags generated by RFTagger (Schmid and Laws, 2008). Similar to **CS2**, We tested the delimited version of normalized words (**CS+Norm2**).

Table 7 shows the results of compound splitting and normalization methods. As a result, normalization on top of compounding did not perform well. Besides, experiments showed that compound word decomposition is crucial and helps vastly to improve translation results 0.43 BLEU points on average over the best system described in Section 2.5.

System	newstest12	newstest11
3LMs	21.78|21.93	20.54|20.68
CS1	22.01|22.21	20.63|20.89
CS2	22.06|22.22	20.74|20.99
CS+Norm2	21.96|22.16	20.70|20.88
CS+Norm1	20.63|20.76	22.01|22.16

Table 7: Compound Splitting Results.

2.7 Average of Weights

As mentioned in Section 2.1, we performed tuning with five different seeds. We averaged the five tuning weights and directly applied these weights during the decoding. Table 8 shows that using the average of several tuning weights performs better than each individual tuning (0.2 BLEU points).

System	newstest12	newstest11
CS2	22.06|22.22	20.74|20.99
Avg. of Weights	22.27	21.07

Table 8: Average of Weights Results.

2.8 Other parameters

In addition to the experiments described in the earlier sections, we removed the *-drop-unknown* parameter which gave us a 0.5 BLEU points improvement. We also included the monotone-at-punctuation, *-mp* in decoding. We handled out-of-vocabulary (OOV) words by lemmatizing the OOV words. Moreover, we added all development data in training after fixing the parameter weights as described in Section 2.7. Although each of these changes increases the translation scores each gave less than 0.1 BLEU point improvement. Table 9 shows the results of the final system after including all of the approaches except the ones described in Section 2.2 and 2.3.

System	newstest12	newstest11
Final System	22.59|22.77	21.86|21.93
Avg. of Weights	22.66	22.00
+ tune data in train	−−	22.09

Table 9: German-to-English Final System Results.

3 English-German

For English-to-German translation system, the baseline setting is the same as described in Section 2.1. We also added the items that showed positive improvement in the German to English SMT system such as using 2 LMs, tuning with five seeds and averaging tuning parameters, using *-mp*, and not using *-drop-unknown*. Table 10 shows the experimental results for English-to-German SMT systems. Similar to the German-to-English direction, tuning with a big development data outperforms the baseline 0.26 BLEU points (on average). Additionally, averaging the tuning weights of different seeds results in 0.2 BLEU points improvement.

System	newstest12	newstest11
Baseline	16.95|17.03	15.93|16.13
+ Big Tune	16.82|17.01	16.22|16.37
Avg. of Weights	16.99	16.47

Table 10: English to German Final System Results.

4 Final System and Results

Table 11 shows our official submission scores for German-English SMT systems submitted to the WMT'13.

System	newstest13
De-En	25.60
En-De	19.28

Table 11: German-English Official Test Submission.

5 Conclusion

In this paper, we described our submissions to WMT'13 Shared Translation Task for German-English language pairs. We used phrase-based systems with a big tuning set which is a combination of the development sets from last four years. We tuned the systems on this big tuning set with five different tunes. We averaged these five tuning weights in the final system. We trained 4-gram language models one from parallel data and one from monolingual data. Moreover, we trained a 4-gram language model with Gigaword v4 for German-to-English direction. For German-to-English, we performed a different compound splitting method instead of the Moses splitter. We obtained a 1.7 BLEU point increase for German-to-English SMT system and a 0.5 BLEU point increase for English-to-German SMT system for the internal test set *newstest2011*. Finally, we submitted our German-to-English SMT system with a BLEU score 25.6 and English-to-German SMT system with a BLEU score 19.3 for the official test set *newstest2013*.

References

Alexandre Allauzen, Josep M. Crego, İlknur Durgar El-Kahlout, and Francois Yvon. 2010. Limsi's statistical translation systems for wmt'10. In *Proceedings of the Fifth Workshop on Statistical Machine Translation*, pages 54–59.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

İlknur Durgar El-Kahlout and Francois Yvon. 2010. The pay-offs of preprocessing German-English statistical machine translation. In *Proceedings of the Seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 251–258.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Proceedings of ACL WSETQANLP*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 181–184.

Philipp Koehn and Kevin Knight. 2003. Emprical methods for compound splitting. In *Proceedings of European Chapter of the ACL (EACL)*, pages 187–194.

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 ACL Demo and Poster Session*, pages 177–180.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of HLT-NAACL*, pages 169–176.

Coşkun Mermer, Murat Saraçlar, and Ruhi Sarkaya. 2013. Improving statistical machine translation using bayesian word alignment and gibbs sampling. *IEEE Transactions on Audio, Speech and Language Processing*, 21:1090–1101.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-granined pos tagging. In *Proceedings of COLING*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 257–286.