Method of Selecting Training Data to Build a Compact and Efficient Translation Model

Keiji Yasuda[†,‡], **Ruiqiang Zhang**[†,‡], **Hirofumi Yamamoto**[†,‡] and **Eiichiro Sumita**[†,‡]
[†]National Institute of Communications Technology
[‡]ATR Spoken Language Translation Research Laboratories
2–2–2, Hikaridai, "Keihanna Science City", Kyoto, 619–0288 Japan
{keiji.yasuda,ruiqiang.zhang}@nict.go.jp
{hirofumi.yamamoto,eiichiro.sumita}@nict.go.jp

Abstract

Target task matched parallel corpora are required for statistical translation model training. However, training corpora sometimes include both target task matched and unmatched sentences. In such a case, training set selection can reduce the size of the translation model. In this paper, we propose a training set selection method for translation model training using linear translation model interpolation and a language model technique. According to the experimental results, the proposed method reduces the translation model size by 50% and improves BLEU score by 1.76% in comparison with a baseline training corpus usage.

1 Introduction

Parallel corpus is one of the most important components in statistical machine translation (SMT), and there are two main factors contributing to its performance. The first is the quality of the parallel corpus, and the second is its quantity.

A parallel corpus that has similar statistical characteristics to the target domain should yield a more efficient translation model. However, domain-mismatched training data might reduce the translation model's performance. A large training corpus obviously produces better quality than a small one. However, increasing the size of the training corpus causes another problem, which is increased computational processing load. This problem not only affects the training of the translation model, but also its applications. The reason for this is that a large amount of training data tends to yield a large translation model and applications then have to deal with this model.

We propose a method of selecting translation pairs as the training set from a training parallel corpus to solve the problem of an expanded translation model with increased training load. This method enables an adequate training set to be selected from a large parallel corpus by using a small in-domain parallel corpus. We can make the translation model compact without degrading performance because this method effectively reduces the size of the set for training the translation model. This compact translation model can outperform a translation model trained on the entire original corpus.

This method is especially effective for domains where it is difficult to enlarge the corpus, such as in spoken language parallel corpora (Kikui et al., 2006). The main approach to recovering from an undersupply of the in-domain corpus has been to use a very large domain-close or out-of-domain parallel corpus for the translation model training (NIST, 2006). In such case, the proposed method effectively reduces the size of the training set and translation model.

Section 2 describes the method of selecting the training set. Section 3 details the experimental results for selecting the training set and actual translation from the International Workshop on Spoken Language Translation 2006 (IWSLT2006). Section 4 compares the results of the proposed method with those of the conventional method. Section 5 concludes the paper.

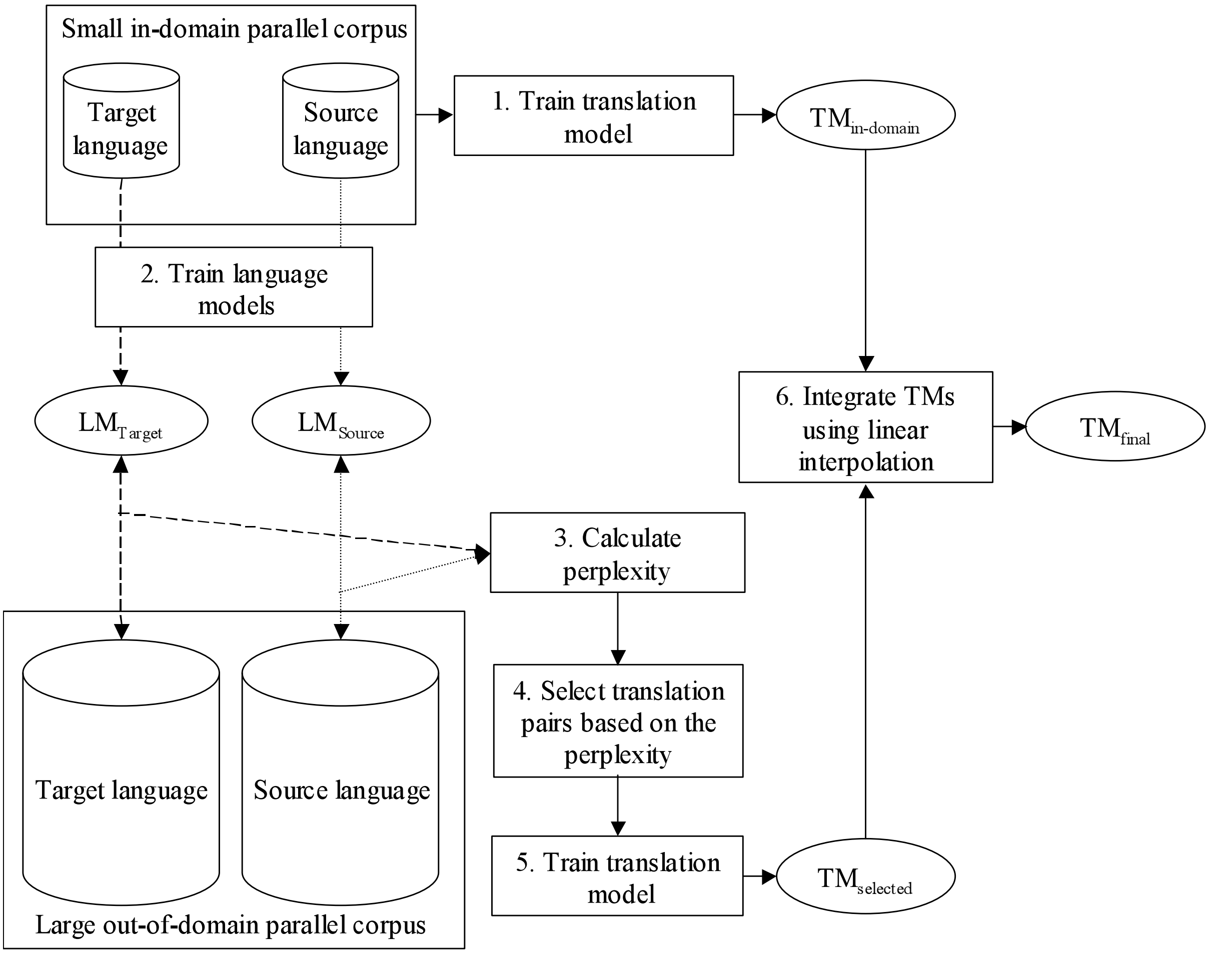
Small in-domain parallel corpus
Target language
Source language
1. Train translation model
TM_{in-domain}
2. Train language models
LM Target
LM Source
6. Integrate TMs using linear interpolation
TM final
3. Calculate perplexity
4. Select translation pairs based on the perplexity
5. Train translation model
TM selected
Target language
Source language
Large out-of-domain parallel corpus

Figure 1: Framework of method.

2 Method

Our method use a small in-domain parallel corpus and a large out-of-domain parallel corpus, and it selects a number of appropriate training translation pairs from the out-of-domain parallel corpus. Figure 1 is a flow diagram of the method. The procedure is as follows:

1. Train a translation model using the in-domain parallel corpus.

2. Train a language model using the source language side or/and target language side of the in-domain corpus.

3. Calculate the word perplexity for each sentence (in source language side or/and target language side) in the out-of-domain corpus by using the following formulas.

$$PP_e = P_e(S_e)^{-\frac{1}{n_e}} \tag{1}$$

where PP_e is the target language side perplexity, and P_e is the probability given by the target side language model. S_e is the target language sentence in the parallel corpus, and n_e is the number of words in the sentence.

We can also calculate the perplexity in the source language (PP_f) in the same way.

$$PP_f = P_f(S_f)^{-\frac{1}{n_f}} \tag{2}$$

If we use perplexities in both languages, we can calculate average perplexity (PP_{e+f}) by using the following formula.

$$PP_{e+f} = (PP_e \times PP_f)^{\frac{1}{2}} \tag{3}$$

Table 1: Size of parallel corpora

	# of sentences		# of words		Explanation
	English	Chinese	English	Chinese	
In-domain parallel corpus	40 K	40 K	320 K	301 K	Basic Travel Expressions Corpus
Out-of-domain parallel corpus	2.5 M	2.5 M	62 M	54 M	LDC corpus (LDC 2002T01, LDC2003T17, LDC2004T07, LDC2004T08, LDC2005T06 and LDC2005T10)

4. Select translation pairs from the out-of-domain parallel corpus. If the perplexity is smaller than the threshold, use translation pairs as the training set. Otherwise, discard the translation pairs.

5. Train a translation model by using the selected translation pairs.

6. Integrate the translation model obtained in 1 and 6 by linear interpolation.

3 Experiments

We carried out statistical machine translation experiments using the translation models obtained with the proposed method.

3.1 Framework of SMT

We employed a log-linear model as a phrase-based statistical machine translation framework. This model expresses the probability of a target-language word sequence (e) of a given source language word sequence (f) given by

$$P(e|f) = \frac{\exp\left(\sum_{i=1}^{M} \lambda_i h_i(e,f)\right)}{\sum_{e'} \exp\left(\sum_{i=1}^{M} \lambda_i h_i(e',f)\right)} \quad (4)$$

where $h_i(e,f)$ is the feature function, λ_i is the feature function's weight, and M is the number of features. We can approximate Eq. 4 by regarding its denominator as constant. The translation results ($\hat{e}$) are then obtained by

$$\hat{e}(f, \lambda_1^M) = \mathrm{argmax}_e \sum_{i=1}^{M} \lambda_i h_i(e,f) \quad (5)$$

3.2 Experimental conditions

3.2.1 Corpus

We used data from the Chinese-to-English translation track of the IWSLT 2006(IWSLT, 2006) for the experiments. The small in-domain parallel corpus was from the IWSLT workshop. This corpus was part of the ATR Bilingual Travel Expression Corpus (ATR-BTEC) (Kikui et al., 2006). The large out-of-domain parallel corpus was from the LDC corpus (LDC, 2007). Details on the data are listed in Table 1. We used the test set of the IWSLT2006 workshop for the evaluation. This test set consisted of 500 Chinese sentences with eight English reference translations per Chinese sentence.

For the statistical machine-translation experiments, we first aligned the bilingual sentences for preprocessing using the Champollion tool (Ma, 2006). We then segmented the Chinese words using Achilles (Zhang et al., 2006). After the segmentation, we removed all punctuation from both English and Chinese corpuses and decapitalized the English corpus. We used the preprocessed data to train the phrase-based translation model by using GIZA++ (Och and Ney, 2003) and the Pharaoh tool kit (Koehn et al., 2003).

3.2.2 Features

We used eight features (Och and Ney, 2003; Koehn et al., 2003) and their weights for the translations.

1. Phrase translation probability from source language to target language (weight = 0.2)

2. Phrase translation probability from target language to source language (weight = 0.2)

3. Lexical weighting probability from source language to target language (weight = 0.2)

4. Lexical weighting probability from source target to language weight = 0.2)

5. Phrase penalty (weight = 0.2)

6. Word penalty (weight $= -1.0$)
7. Distortion weight (weight = 0.5)
8. Target language model probability (weight = 0.5)

According to a previous study, the minimum error rate training (MERT) (Och, 2003), which is the optimization of feature weights by maximizing the BLEU score on the development set, can improve the performance of a system. However, the range of improvement is not stable because the MERT algorithm uses random numbers while searching for the optimum weights. As previously mentioned, we used fixed weights instead of weights optimized by MERT to remove its unstable effects and simplify the evaluation.

3.2.3 Linear interpolation of translation models

The experiments used four features (Feature # 1 to 4 in 3.2.2) as targets for integration. For each feature, we applied linear interpolation by using the following formula.

$$h(e, f) = \mu_{out} h_{out}(e, f) + (1 - \mu_{out}) h_{in}(e, f) \quad (6)$$

Here, $h_{in}(e, f)$ and $h_{out}(e, f)$ are features trained on the in-domain parallel corpus and out-of-domain corpus, respectively. μ_{out} is the weight for the feature trained on the out-of-domain parallel corpus.

3.2.4 Language model

We used a Good-Turing (Good, 1953) 3-gram language model for data selection.

For the actual translation, we used a modified Kneser-Ney (Chen and Goodman, 1998) 3-gram language model because modified Kneser-Ney smoothing tended to perform better than the Good-Turing language model in this translation task. For training of the language model, only the English side of the in-domain corpus was used. We used the same language model for the entire translation experiment.

3.3 Experimental results

3.3.1 Translation performance

Figure 2 and 3 plot the results of the experiments. The horizontal axis represents the weight for the out-of-domain translation model, and the vertical axis represents the automatic metric of translation quality (BLEU score (Papineni et al., 2002) in Fig. 2, and NIST score (NIST, 2002) in Fig. 3). Thick straight broken lines in the figures indicate automatic scores of a baseline system. This base line system was trained on the in-domain and all of the out-of-domain corpus (2.5M sentence pairs). These data were concatenated before training; then one model was trained without linear interpolation. The five symbols in the figures represent the sizes (# of sentence pairs) of the selected parallel corpus. Here, the selection was carried out by using Eq. 1. For automatic evaluation, we used the reference translation with a case unsensitive and no-punctuation setting. Hence, higher automatic scores indicate better translations; the selected corpus size of 1.2M (×) indicates the best translation quality in Fig. 2 at the point where the weight for the out-of-domain translation model is 0.7.

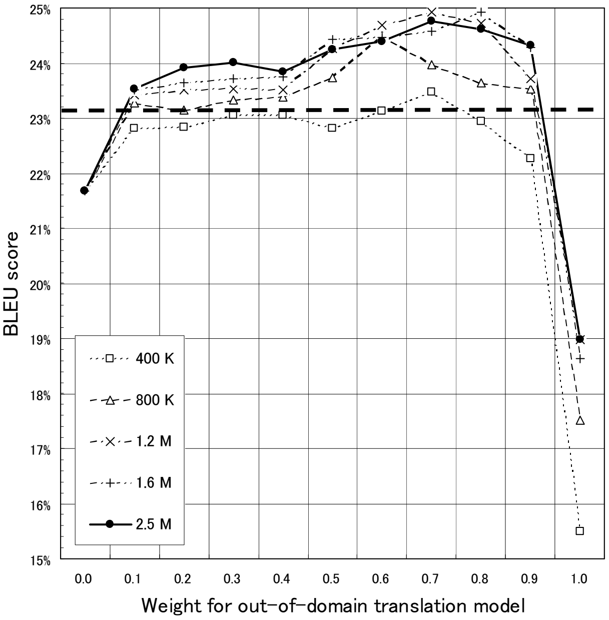

Figure 2: Results of data selection and linear interpolation (BLEU)

In contrast to Fig. 2, Fig. 3 shows no improvements to the NIST score by using the baseline out-of-domain usage. The optimal weights for each corpus size are different from those in Fig. 2. However, there is no difference in optimal corpus size; i.e., the selected corpus size of 1.2M gives the best NIST score.

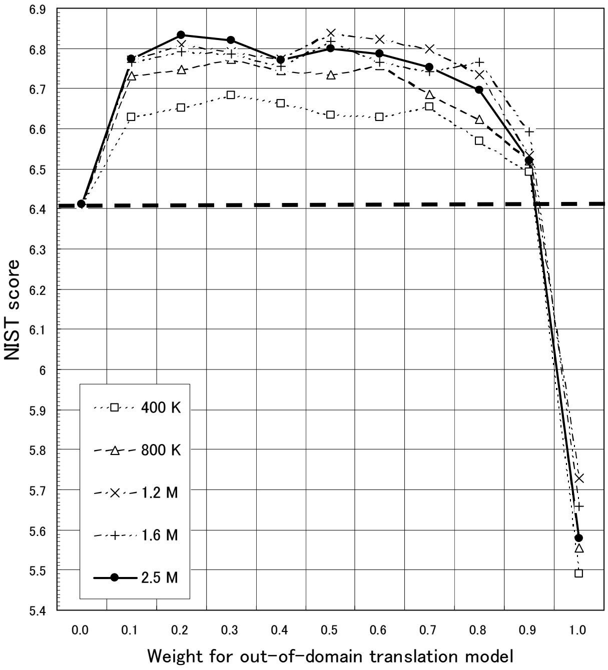

Figure 3: Results of data selection and linear interpolation (BLEU)

Table 2: Size of integrated phrase tables

Corpus size (Sentence pairs)		Size of phrase table (Bytes)
In-domain	Out-of-domain	
40 K	0	14 M
40 K	1.2 M	917 M
40 K	2.5 M	1.8 G

3.3.2 Size of the translation models

Table 2 lists the sizes of the translation models of the baseline and optimum-size training corpus. The size of the phrase table is the uncompressed file size of the phrase table trained by the Pharaoh tool kit. As the table indicates, our method reduced the model sizes by 50%.

This reduction had a positive effect on the computational load of decoding.

3.3.3 Equations for the selection

The experiments described above used only target language side information, i.e., Eq. 1, for the data selection. Here, we compare selection performances of Eqs. 1, 2, and 3. Table 3 shows the results. The first row shows the results of using only the in-domain parallel corpus. The second row shows results of the baseline. The third row shows the results of using linear interpolation without data selection. Comparing the results for the three equations, we see that Eq. 1 gives the best performance. It outperforms not only the baseline but also the results obtained by using all of the (2.5M) out-of-domain data and linear interpolation.

The results of using source language side information (Eq. 2) and information from both language sides (Eq. 3) still showed better performance than the baseline system did.

4 Comparison with conventional method

There are few studies on data selection for translation model training. Most successful and recent study was that of (Lu et al., 2007). They applied the TF*IDF framework to translation model training corpus selection. According to their study, they obtained a 28% translation model size reduction (A 2.41G byte model was reduced to a 1.74G byte model) and 1% BLEU score improvement (BLEU score increased from 23.63% to 24.63%). Although there results are not directly comparable to ours [??] because of the differences in the experimental setting, our method outperforms theirs for both aspects of model size reduction and translation performance improvement (50% model size reduction and 1.76% BLEU score improvement).

5 Conclusions

We proposed a method of selecting training sets for training translation models that dramatically reduces the sizes of the training set and translation models.

We carried out experiments using data from the Chinese-to-English translation track of the IWSLT evaluation campaign. The experimental results indicated that our method reduced the size of the training set by 48%. The obtained translation models were half the size of the baseline.

The proposed method also had good translation performance. Our experimental results demonstrated that an SMT system with a half-size translation model obtained with our method improved the BLEU score by 1.76%. (Linear interpolation improved BLEU score by 1.61% and data selection improved BLEU score by an additional 0.15%.)

Table 3: Results of data selection by using Eqs. 1, 2, and 3

Corpus size (Sentence pairs)		Selection method	Optimal weight for out-of-domain model	BLEU score
In-domain	Out-of-domain			
40 K	0	N/A	N/A	21.68%
40 K	2.5 M	N/A	N/A	23.16%
40 K	2.5 M	N/A	0.7	24.77%
40 K	1.2 M	Eq. 1	0.7	24.92%
40 K	1.2 M	Eq. 2	0.8	24.76%
40 K	1.2 M	Eq. 3	0.6	24.56%

We also compared the selections using source language side information, target language side information and information from both language sides. The experimental results show that target language side information gives the best performance in the experimental setting. However, there are no large differences among the different selection results. The results are encouraging because they show that the in-domain mono-lingual corpus is sufficient to select training data from the out-of-domain parallel corpus.

References

S. F. Chen and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. In *Technical report TR-10-98, Center for Research in Computing Technology (Harvard University).*

I. J Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3):237–264.

IWSLT. 2006. IWSLT: International Workshop on Spoken Language Translation. http://www.slc.atr.jp/IWSLT2006/.

G. Kikui, S. Yamamoto, T. Takezawa, and E. Sumita. 2006. Comparative study on corpora for speech translation. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 14(5), pages 1674–1682.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proc. of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 127–133.

LDC. 2007. Linguistic Data Consortium. http://www.ldc.upenn.edu/.

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350.

X Ma. 2006. Champollion: A Robust Parallel Text Sentence Aligner. In *Proc. of international conference on Language Resources and Evaluation (LREC)*, pages 489–492.

NIST. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurence Statistics. http://www.nist.gov/speech/tests/mt/mt2001/resource/.

NIST. 2006. The 2006 NIST Machine Translation Evaluation Plan (MT06). http://www.nist.gov/speech/tests/mt/doc/mt06_evalplan.v3.pdf.

F. J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

F. J. Och. 2003. Minimum Error Rate Training for Statistical Machine Translation. *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

R. Zhang, G. Kikui, and E. Sumita. 2006. Subword-based Tagging by Conditional Random Fields for Chinese Word Segmentation. *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Short Paper:193–196.