# Approach to Selecting Best Development Set for Phrase-Based Statistical Machine Translation*

Peng Liu, Yu Zhou, and Chengqing Zong

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
Room 1010, 95 Zhongguancun East Road, Beijing 100190, China
{pliu, yzhou, cqzong}@nlpr.ia.ac.cn

**Abstract.** In phrase-based statistical machine translation system, the parameters of model are usually obtained from minimum error rate training (MERT) on the development set. So the development set has a great influence on the performance of the translation system. Generally, more development set will achieve more effective and robust parameters but consume much more time on MERT process. In this paper, we propose two methods to select sentences from the large development set, based on the phrase and the sentence structure respectively. The experimental results show that our methods can get better translation performance than the baseline system on the compact development set by using a state-of-the-art SMT system.

**Keywords:** Development set selecting, Domain adaption, Phrase-based statistical machine translation, Similarity measure

## 1 Introduction

In recent years, the phrase-based statistical machine translation model obtains more attention and achieves good translation performance. But the quality of the translation results is greatly influenced by the model's parameters. How to get a group of parameters which can make good translation results becomes a problem which we focus on. In general, we could get the optimized parameters though minimum error rate training (MERT) on the development set (Och, 2003). The MERT will continue running until the BLEU score on the development set convergences and then we translate the test set to the target language using this group of parameters.

Since we get the optimized parameters on the development set, it becomes an important factor to the quality of the final translation results. Usually, we run the MERT on all of the development set, but when the development set is in large-scale and there are many long sentences included in it, the MERT will consume too long time on translation and parameters' adjusting. Nevertheless we can not sure whether the parameters trained on it are optimal. So what we are interested in are: 1) How many sentences are adequate for the MERT and what kind of sentences contribute more to the MERT? 2) How can we select such development set to obtain more effective and robust parameters with less time and without performance losing?

Based on the analysis above, our aim is to find a method to select sentences from the whole development set, and use the new development set we can save time on MERT process as well as improve the performance of the translation system. In intuition, if the sentences in the

* The research work has been partially funded by the Natural Science Foundation of China under grant No.60736014, 60723005 and 90820303, the National Key Technology R&D Program under grant No. 2006BAH03B02, the Hi-Tech Research and Development Program (863 Program) of China under grant No. 2006AA010108-4, and also supported by the China-Singapore Institute of Digital Media as well.

development set are more similar to the ones in the test set, the parameters trained on them will be more appropriate for the test set. So the intuitive way is to select the part of sentences from the whole development set base on some similarity measures. There are many similarity measure have been proposed by the former researchers, so which measure is suit for our task is the key problem we should focus on.

The remainder of the paper is organized as follows. Section 2 will discuss the related work. We present our methods in Section 3 and discuss the experimental results in Section 4. Finally, we give our conclusions in Section 5.

## 2 Related Work

In fact, our aim is somewhat like domain adaptation. The task of domain adaptation is to develop learning algorithms that can be easily ported from one domain to another (Daumé III, 2007). What we want to do is selecting appropriate sentences from the development set, on which the MERT will train a group of parameters that can make the translation results have the best quality. The initial development set is like the source domain, and the test set is like the target domain. We have a lot of data in source domain, and desire a model that performs well in the target domain. Wu *et al.* (2008) trained a baseline system using out-of-domain corpora and then used in-domain resource to improve the in-domain performance. The development set is composed of in-domain data and out-of-domain data. We select the sentences which are domain-matched and use them to get optimal parameters.

The concept of similarity is very important in NLP applications, such as the information extraction, information retrieval and document clustering. There are many similarity measures have been proposed by former researchers. Bergsma and Kondrak (2007) proposed an alignment-based discriminative framework for string similarity. They gathered features from substring pairs consistent with a character-based alignment of the two strings. Li *et al.* (2006) presented an algorithm which takes account of semantic information and word order information implied in the sentence to calculate the similarity between very short texts of sentence length. Budanitsky and Hirst (2006) evaluated five of WordNet based semantic measures, by comparing their performance in detecting and correcting real-word spelling errors.

However, there are some differences between our task and the common similarity measures. If we can find similar sentences for each sentence in the test set, we are sure that the parameters trained on them can improve the performance significantly. However, in the limited scale of the development set, the sparse data problem becomes too serious to find a similar sentence for each sentence in the test set. The normal string or semantic similarity measures are not suit for our task.

Some researchers tried to optimize the training set to improve the performance of translation system. Yasuda *et al.* (2008) selected the training set for translation model training using linear translation model interpolation and a language model technique. They focused on the training set, and they used the fixed weights in the translation. On the contrary, we focus on the development set, and use parameters optimized by MERT. Matsoukas *et al.* (2009) proposed a discriminative training method to assign a weight for each sentence in the training set. In this way, they limited the negative effects of low quality training data. We pay more attention to the parameter estimation, and eager to get a group of optimized parameters for the translation model.

## 3 Criterion for Selecting Development Data

As mentioned above, the normal similarity measures on sentence level are not easy to be applied for our task because of the sparse data problem. In order to achieve our purpose, we need to select some sentences which are similar to the entire test set. We establish a new development set using these sentences and hope to get better performance on it.

The framework of our methods is shown in Figure 1. The TST is the test set, the DEV is the development set, and the New DEV is the development set we extract from the DEV base on some similarity measures. We build criteria from the TST, assign different weight to the basic unit. Then we use the weight to measure the sentences in the DEV by giving each sentence a score. At last, we select the New DEV based on the score.

There are two basic methods to calculate the similarity. One is based on the surface features, such as the words and phrases; the other is based on some deeper features, such as the sentences structure. We propose similarity measures to select the sentences in these two ways respectively.

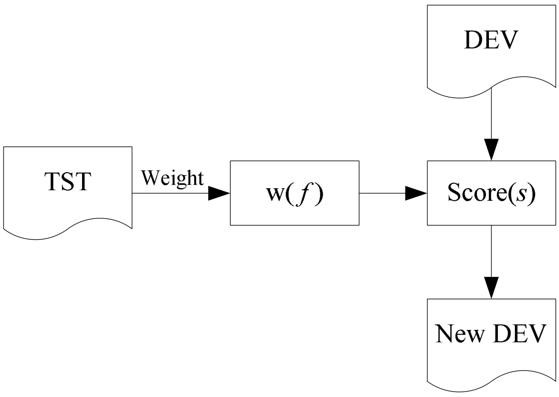

**Figure 1:** Framework of selecting development set

## 3.1 Phrase-based Method

For the phrase-based statistical machine translation model, the basic translate unit is phrase, that is to say, a continuous word sequence (Koehn *et al.*, 2003). It is a natural idea that using the phrase to measure the similarity between the test set and the development set. If the sentences which we selected contain more phrases in the test set, the sentences are more similar to the test set. Then we try to select sentences from the development set which can cover more phrases of the test set corpus.

In this method, the phrases in test set play a vital role. So, firstly we extract all the phrases from the test set and assign them different weights. We take two aspects into account to estimate the weight of phrase: the information it contained and the length of the phrase. In information theory (Cover and Thomas, 1991; Lin, 1998), the information contained in a statement is measured by the negative logarithm of the probability of the statement. So we should estimate the probability of each phrase first. We class the phrases with their lengths and only use the phrases which length is not longer than four in order to avoid the sparse data problem. We calculate the probabilities of the phrases based on their lengths respectively. For a phrase $f$, its length $|f|$ is $n$ and the probability $p(f)$ is estimated by following formula:

$$p(f)=\frac{count(f)}{\sum_{|f_i|=n} count(f_i)} \qquad (1)$$

Where the numerator $count(f)$ is the total number of phrase $f$ appears in the test set, and the denominator is the total number of the phrases which length is equal to $n$. Then the information contained in phrase $f$ is calculate by formula (2),

$$I(f) = -\log p(f) \qquad (2)$$

In this way, we get the information contained in each phrase. Because the translation model is based on phrase, the longer phrase will lead to better translation. So we take $n$, the length of phrase, into account. We use the square root of length, but not the length directly because of the data smoothing. And the formula to calculate the weight of each phrase is shown below.

$$w(f) = \sqrt{n} \cdot I(f) \qquad (3)$$

Now, we get the weight for each phrase in the test set base on the length of the phrase and the information it contains. Then we can estimate the weight of sentence in the development set by the phrase weight. For a sentence $s$ in the development set, if more phrases it contains appear in the test set, we assign it a larger score. The score of the sentence is calculated by the following formula:

$$Score(s) = \sum w(f) \qquad (4)$$

We extract all the phrases whose length is not longer than four in sentence $s$, and we add all the weights of phrases together. If a phrase does not appear in the test set, the weight of the phrase is set to zero. The sentences are sorted by their score in a descending order. We choose higher score sentences to combine new development set. We run MERT on different scale development set to get system parameters. At last, the test set is translated using this group of parameters, and the results will be presented in Section 4.

### 3.2 Structure-based Method

The phrase-based method only uses phrase, a surface feature, to estimate the weight of sentences. And it doesn't contain any deep features, such as the sentence structure. So we try to use some of deep features to help choosing the development set.

As we do in the phrase-based method, we want to find sentences which have the structures can cover the great mass of the test set. So we firstly parse all the source language sentences (including the development set and the test set) using Stanford lexicalized parser version 1.6[1]. Each sentence is parsed into a phrase-structure tree. If we use the entire tree to calculate similarity, the sparse data will become an insuperable problem. So we use the subtree of the phrase-structure tree as the basic unit. We only use the structure information, so the subtree doesn't contain any word. The range of subtree depth is limited to from two to four in order to avoid the sparse data problem. An example of phrase-structure tree and subtrees is shown in Figure 2.

We consider two aspects to estimate the weight of the subtree: depth and information, as same as the phrased-based method. For a subtree $t$, and its depth is $d$, let's assume its probability is $p(t)$, which is estimated from the test set, and the information contained in it is calculate by formula (5).

[1] http://nlp.stanford.edu/software/lex-parser.shtml

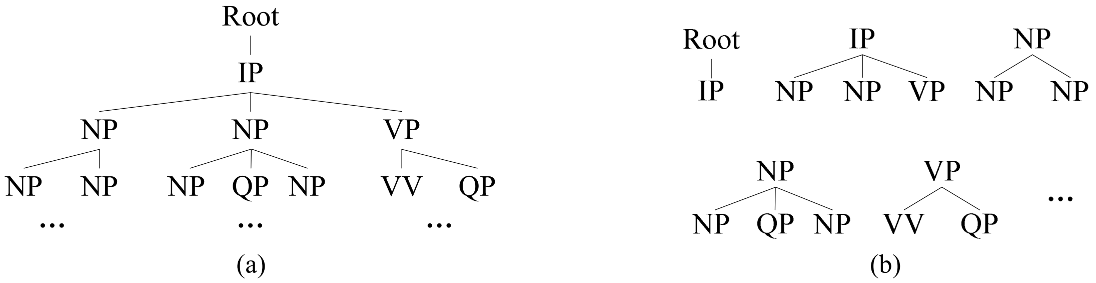

**Figure 2:** (a) a phrase-structure tree; (b) subtrees which depths are 2 contained in the phrase-structure tree.

$$I(t) = -\log p(t) \tag{5}$$

Then the weight of each subtree in the test set is calculated by the following formula.

$$w(t) = \sqrt{d} \cdot I(t) \tag{6}$$

Now we can measure the weight of each sentence in the development set by all the subtrees it contained. If a subtree does not appear in the test set, the weight of the subtree is assigned to zero. For a sentence $s$ in the development set, the score is calculated by following formula:

$$Score(s) = \sum w(t) \tag{7}$$

Then we sort the sentences according to their scores, and select the sentences in a descending order. We run MERT on different scale of development set, and using the parameters to translate the test set. The experimental results are presented in Section 4.

## 4 Experiments and Results

We did our experiments on Chinese to English translation task. The translation model is built on NIST corpus, using about 300,000 sentences to extract phrases and build language model. And we use newswire portion of NIST MT05, MT06 and MT08 test set as our development set, and use CWMT 2008 (China Workshop on Machine Translation)[2] test set as our test set. The development set has 4103 sentences, each one has four reference translations; and the test set has 1006 sentences. We use MOSES[3] as our translation system.

### 4.1 Baseline

Before the experimental results are presented, we wonder what kind of performance we will get if we only use a part of the development set in their initial order. So we use a part of the development set in their initial order, and run MERT on them to get translation system parameters. After the MERT finished, we use the parameters to translate the test set. And we use the results as our baseline. The experimental results are presented in Figure 3.

In Figure 3, the $x$ axis is the ratio of the total number of development set sentences divides the total number of the sentences in the test set. The left vertical axis is the BLEU score (BLEU-4) and the right vertical axis is the average sentence length of the development. The square point is the BLEU score on the test set; the triangle point is the average sentence length of the new development set.

[2] http://nlpr-web.ia.ac.cn/cwmt-2008/

[3] http://www.statmt.org/moses/

When the scale of the development set is small, the quantity of sentences has a great influence on the performance of SMT system. The BLEU score of the translation results continue rising with the increasing of the development set. When $x$ is equal to 0.5, i.e. the development set is only half of the test set, the BLEU score is only 0.0909. When the quantity of the development set increases to as many as the test set, the BLEU score rapidly increase to 0.1359. Then the performance continues improving when we add more sentences to the development set. When the development set is three times as many as the test set, the BLEU has a little decline. When we add more sentences to the development set, the performance is relatively stable and only a little change occurred.

From Figure 3, it is clear that the performance of the translation system is greatly influenced by the development set when the development set is in small scale. However, when the development set increase to a special scale, the performance will keep stable. Adding more sentences to the development set will not increase the performance evidently but consume more time on translation and adjusting parameters.

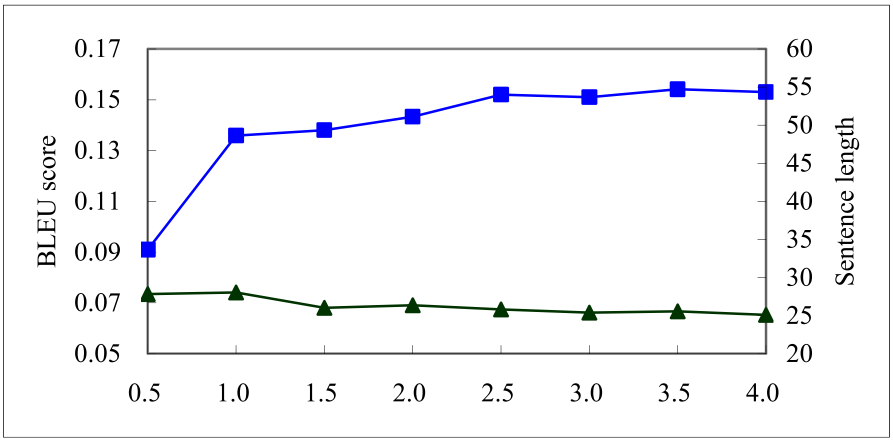

**Figure 3:** Results of baseline system

Because we select the sentences in their initial order without any other measures, the average sentence length does not change greatly. The value is between 25.10 when $x$ is equal to 4.0 and 28.03 when $x$ is equal to 1.0.

## 4.2 Results of Phrase-based Method

We select development set using two methods we proposed above, and run MERT on them to get a group of optimal parameters. Then we translate the test set using these parameters. The experimental results are shown in Table 1. The first row is the ratio which the total number of development set divides to the total number of test set. The second row is the results of baseline system, selecting sentences in their initial order. The third row is the results of phrase-based method; and the last row is the results of structure-based method. The last column is the average value of each experiment.

The average sentence length is shown in Table 2. The first row is as same as the Table 1. And the last three rows are the average sentence length of each development set. We also present the experimental results of phrase-based methods in Figure 4. The horizontal axis and vertical axis are as same as Figure 3. The square point is the BLEU score of the test set; and the triangle point is the average length of the development set sentences. The recall ratio of the phrase is presented in Table 3. The first row is as same as Table 1; the second to the fifth rows are the recall ratio of phases with different length; and the last row is the average value of four kinds of recall ratio.

**Table 1:** BLEU score of experiment results

| Ratio | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.0909 | 0.1359 | 0.1380 | 0.1433 | 0.1520 | 0.1510 | **0.1541** | 0.1530 | 0.1398 |
| Phrase | 0.1445 | 0.1519 | 0.1538 | **0.1572** | 0.1546 | 0.1545 | 0.1556 | 0.1535 | **0.1532** |
| Structure | 0.1397 | 0.1536 | 0.1518 | 0.1523 | **0.1587** | 0.1550 | 0.1576 | 0.1554 | 0.1530 |

**Table 2:** The average length of the development set sentences

| Ratio | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 |
|---|---|---|---|---|---|---|---|---|
| Baseline | 27.83 | 28.03 | 26.03 | 26.34 | 25.80 | 25.38 | 25.53 | 25.10 |
| Phrase | 50.27 | 43.46 | 39.22 | 35.83 | 32.88 | 30.30 | 27.88 | 25.50 |
| Structure | 49.69 | 43.30 | 39.08 | 35.65 | 32.66 | 30.11 | 27.79 | 25.45 |

**Table 3:** The recall ratio of the phrase-based method

| Ratio | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 |
|---|---|---|---|---|---|---|---|---|
| Length=1 | 0.687 | 0.821 | 0.890 | 0.931 | 0.960 | 0.979 | 0.992 | 1.000 |
| Length=2 | 0.545 | 0.728 | 0.825 | 0.898 | 0.940 | 0.973 | 0.993 | 1.000 |
| Length=3 | 0.528 | 0.718 | 0.839 | 0.914 | 0.974 | 0.995 | 1.000 | 1.000 |
| Length=4 | 0.646 | 0.826 | 0.910 | 0.951 | 0.993 | 1.000 | 1.000 | 1.000 |
| Avg. | 0.601 | 0.773 | 0.866 | 0.923 | 0.967 | 0.987 | 0.996 | 1.000 |

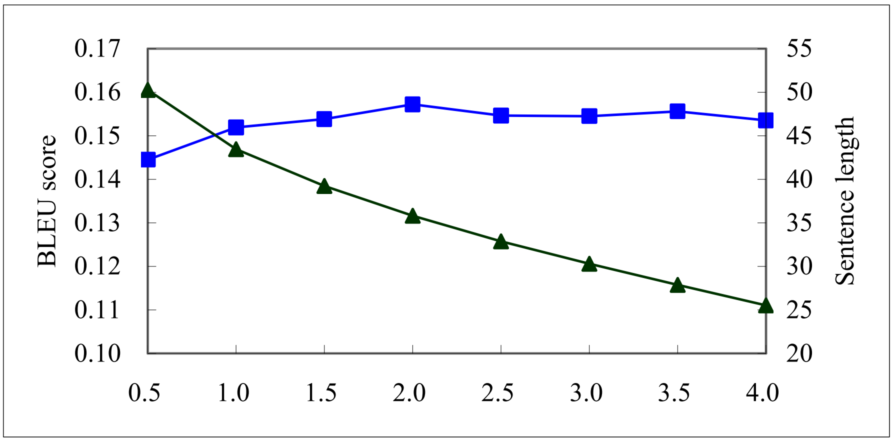

**Figure 4:** Results of phrase-based method

From Figure 4, it is clear that using the phrase-based method, the performance of system get a great enhancement, especially when the development set is in small scale. When $x$ is equal to 0.5, the BLEU score is 0.1445, there is 5.36% BLEU score improvement compare to the baseline method. And with the increasing of development set, the performance continues improving. It reaches the maximum value 0.1572 when $x$ is equal to 2.0, with average phrase recall of 92.3%. Now, most of the common phrases have been covered by the new development set. Then the BLEU score has a litter drop but almost keep stable. From now on, adding more sentences to the development will give little increase for the recall. Comparing to the baseline method, the phrase-based method reaches the maximum value more quickly, using only two times as many as test set sentences. But the baseline method uses 3.5 times as many as the test set. This will consume more time on training process. The baseline method takes 59 min for each iteration in the MERT process to reach the best performance, while the phrased-based method only takes 45 min, saving time 23.7%. When use the same quantity of development set, the performance of phrase-based method is always higher than the baseline system. The average score of phrase-based method is 0.1532, and the average score of baseline system is 0.1398. The former method is 1.34% BLEU score higher than the latter method.

In this method, the average length of the development set sentence decreases monotonically. It drops from 50.27 to 25.50 with the increasing of development sentences. Our method is apt to choose longer sentences. It is easy to understand that longer sentence contains more phrases, and easy to get higher score.

## 4.3 Results of Structure-based Method

The structure-based method experiment results are shown in Figure 5 and the recall ratio are presented in Table 4. From Figure 5, it is clear that the results have similar trend with the phrase-based method but not so stable. The system performs better than the baseline when the quantity is small. When $x$ is 0.5, the BLEU score is 0.1397, 4.88% BLEU score higher than the baseline system. The corresponding recall ratio is 66.2%, less than two third of the common subtree is been covered. And it reaches the maximum value 0.1587 when $x$ is equal to 2.5, 0.67% BLEU score higher than the baseline. And the recall ratio rises to 94.7%. It takes 49 min for each iteration, saving time 16.9%. The average score of the structure-based method is 0.1530, 1.32% higher than the baseline method. But the BLEU score's fluctuation range of this method is larger than the phrase-based method. This makes the average score of the structure-based method is 0.02% lower than the phrase-based method.

**Table 4:** The recall ratio of the structure-based method

| Ratio | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 |
|---|---|---|---|---|---|---|---|---|
| Depth=2 | 0.750 | 0.864 | 0.910 | 0.937 | 0.966 | 0.976 | 0.994 | 0.999 |
| Depth=3 | 0.642 | 0.785 | 0.861 | 0.906 | 0.939 | 0.971 | 0.992 | 1.000 |
| Depth=4 | 0.593 | 0.745 | 0.840 | 0.899 | 0.937 | 0.973 | 0.991 | 0.999 |
| Avg. | 0.662 | 0.798 | 0.870 | 0.914 | 0.947 | 0.973 | 0.992 | 0.999 |

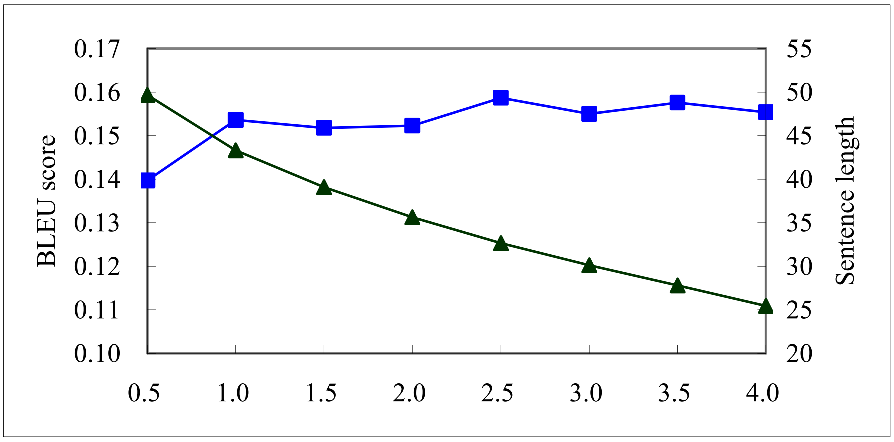

**Figure 5:** Results of structure-based method

Comparing to the phrase-based method, the structure-based method is not so stable. This may because the structure-based method is calculated based on the parsing results. And it is well known that the syntactic analysis is a very difficult problem, and the precision of parser is not high. The F1 score (harmonic mean of precision and recall) of the parser is only about 80%

(Levy and Manning, 2003). So there are many errors in the parsing results. This make the structure-based method is not as stable as the phrase-based method.

The structure-based method is also prior to the longer sentences. Because longer sentence contains more subtrees, and this makes the sentence get a higher score. In Figure 5, the average sentence length decreases monotonically, drops from 49.69 to 25.45, it is almost as same as the phrase-based method.

## 5 Conclusions

From the experimental results, we can get some useful conclusions:

First, the scale of development set is not the larger the better. The BLEU score on the test set will rapidly increase with the increasing of development set scale. However, the increase will become slower if we continue adding sentences to the development set. And the performance keeps stable when the development set is adequate, as shown in all experiments. When we add more sentences to the development set, the BLEU score will not continue increasing but consuming more training time. According to the experimental results, we can get optimal parameters when the development set is at least two times as many as the test set.

Second, the methods we proposed to select development sentences are effective. On each scale of development set, the parameters trained on it will get better performance than the baseline system. Especially when the development set is less than two times as many as the test set. Comparing the baseline and the phrase-based metric, when the development set is 0.5, 1.0, 1.5 and 2.0 times as many as the test set, the BLEU score improve 58.9%, 11.7%, 11.4% and 9.6% respectively. After the development set is twice larger than the test set, the scores are still higher than the baseline. The structure-based method has the same trend with the phrased based method.

Finally, the phrase-based method is powerful than the structure-based method. Two methods have the same trend, but the phrase-based method is more stable. The structure-based method is influenced by the parser errors. From the results, the average BLEU score on phrase-based method is 0.1532, while the average score on structure-based method is 0.1530. The phrase-based method is a little higher than the structure-based method. So both methods are powerful than the baseline system, but the phrase-based method is more effective.

## References

Bergsma, S. and G. Kondrak. 2007. Alignment-Based Discriminative String Similarity. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 656-663, Prague, Czech Republic.

Budanitsky, A. and G. Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1), 13-47.

Cover, T. M. and J. A. Thomas. 1991. *Elements of Information Theory*. New York: Wiley.

Daumé III, H. 2007. Frustratingly Easy Domain Adaptation. *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 256-263.

Koehn, P., F. J. Och and D. Marcu, 2003. Statistical Phrase-Based Translation. *Proceedings of NAACL*, pp. 48-54, Edmonton, Canada.

Levy, R. and C. D. Manning. 2003. Is it Harder to Parse Chinese, or the Chinese Treebank? *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 439-446, Sapporo, Japan

Li, Y., D. McLean, Z. A. Bandar, J. D. O'Shea and K. Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8), 1138-1150.

Lin, D., 1998. An Information-Theoretic Definition of Similarity. *Proceedings of the 5th International Conference on Machine Learning*, pp. 296-304, Madison, Wisconsin.

Matsoukas, S., A.-V. I. Rosti and B. Zhang. 2009. Discriminative Corpus Weight Estimation for Machine Translation. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 708-717, Singapore, Association for Computational Linguistics.

Och, F. J. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 160-167, Sapporo, Japan.

Wu, H., H. Wang and C. Zong, 2008. Domain Adaptation for Statistical Machine Translation with Domain Dictionary and Monolingual Corpora. *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 993-1000, Manchester, UK.

Yasuda, K., R. Zhang, H. Yamamoto and E. Sumita. 2008. Method of Selecting Training Data to Build a Compact and Efficient Translation Model. *Proceedings of the Third International Joint Conference on Natural Language Processing*, pp. 655-660, Hyderabad, India.