International Journal of Computer Processing Of Languages
Vol. 23, No. 4 (2011) 327–348

DOI: 10.1142/S1793840611002395

Approaches to Improving Corpus Quality for Statistical Machine Translation

YU ZHOU*, PENG LIU† AND CHENG ZONG‡

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
*yzhou@nlpr.ia.ac.cn
†pliu@nlpr.ia.ac.cn
‡cqzong@nlpr.ia.ac.cn

The performance of a statistical machine translation (SMT) system depends heavily on the quantity and quality of the bilingual language resource. However, previous work mainly focuses on the quantity and tries to collect more bilingual data. In this paper, to optimize the bilingual corpus to improve the performance of the translation system, we propose some approaches to processing the training corpus by filtering noise and selecting more informative sentences from the training corpus. Also, to coordinate the parameter turning using minimum error rate training (MERT) approach, we propose two methods to select sentences from the large development data which are based on the phrase and sentence structure respectively. Different from the existing methods, our methods do not need so many development data but still obtains effective and robust parameters, while expending little time in the MERT process. The experimental results show that our methods can get better translation performance both in translation quality and speed.

Keywords: Training data selection; development set selection; noise filtering; corpus optimization; statistical machine translation.

1. Introduction

In recent years, the statistical machine translation (SMT) model has attracted more and more attention and achieves competitive translation performance compared to other translation methods. The main merit of the SMT model lies in the fact that it can learn more translation knowledge from a very large scale of training data automatically. However, the performance of an SMT system heavily depends on the quantity and quality of the training data and development set,

because the translation model and language model are built on the training data and the parameters are optimized on the development set through minimum error rate training (MERT) method.

Since the final translation result is greatly influenced by the quantity of the training data and development set, most researchers agree with the opinion: "the more, the better". A lot of previous work mainly focuses on improving the quantity of training data and collecting more bilingual data. Certainly, based on more training data and development data, the probabilities and parameters can become more accurate and lead to better performance. However, a lot of experiments have shown that this is not a wise way to use all the training data for a test data limited on a predefined field [1, 2]. It is obvious that the performance will not be so good by only increasing the training data if there is a big divergence between the training data and test data. In addition, if the quality of bilingual data is poor, the translation performance will be undermined due to the noise caused by the poor data. Moreover, more training data will take more computational resources. It will greatly decrease the translation speed for the great complexity in decoding stage. This will be a hinderance in a practical SMT system development, especially on some mobile devices because they are limited in their CPU capability and memory storage. So a compact and efficient corpus is needed. Therefore, in this paper, we propose some methods to process the bilingual language data by filtering noise and selecting more informative sentences from the training and development corpus.

For processing the training data, a key issue is how to filter the noise — the wrongly aligned sentence pairs, because the noise will cause wrong word-alignments and reduce the performance of the translation results. The other key issue is how to fix the size of the training data. However, for the development data, the size is much smaller than the training data and the noise can be ignored. The main problem is how to select the most informative sentences to tune the translation parameters. If the development set is in large-scale and there are many long sentences included in it, MERT will consume too much time on translation and parameters' adjusting. Nevertheless we can not be sure whether the parameters trained on it are optimal. So what we are interested in are: ① How many sentences are adequate for 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?

In this paper, we present our approaches to filtering the noise in the training data using the length-ratio-based and translation-ratio-based methods. And we estimate the weight of a sentence based on the phrases contained in the sentence. The compact training corpus is built according to the sentence weight. For the

development data, we select sentences based on surface feature and deep feature on phrase level and structure level separately. In addition, we also verify the relationship between the size and the translation performance.

The remainder of this paper is organized as follows. Related work is presented in Section 2. The data optimization methods for selecting training corpus and development corpus are respectively described in Section 3 and Section 4. We give the experimental results in Section 5 and come to the conclusions in Section 6.

2. Related Work

Considering how data scale can greatly influence the translation performance, many researchers have focused on the data collection of training data. They tried to get more parallel data. Resnik and Smith extracted parallel sentences from web resources [3]; Snover *et al*. improved the translation performance using comparable corpora [4]. Nowadays, many researchers have realized that the quality of corpus also plays an important role in translation performance. Therefore, many methods to optimize the parallel data have been emerged and the topic on training data selection has attracted much attention in SMT research. For example, Eck *et al*. proposed an approach to selecting informative sentences based on *n*-gram coverage [5]. They employed the previous unseen *n*-grams contained in a sentence to measure the importance of the sentence. However, they only considered the quantity of the unseen *n*-grams and didn't take the weight of *n*-gram into account. Lü *et al*. proposed a method to select training corpus by information retrieval method [6]. They assumed that the target test data was known before building the translation model and selected the sentences similar to the test data using TF-IDF measure. The limitation of this method is that the test data must be known first, but generally, it is not practical. Yasuda *et al*. used the perplexity as the measure to select parallel translation pairs from the out-of-domain corpus. They integrated the translation model by using linear interpolation method [7]. Matsoukas *et al*. proposed a discriminative training method to assign a weight for each sentence in the training set [8]. Their purpose is to restrict the negative effects of training data with low quality.

For the development data selection, inspired by the ideas of domain adaptation (DA), some researchers treat the problem to select development set for SMT as a problem of DA. The task of DA is to develop proper learning algorithms that can be easily ported from one domain to others. Therefore, some researchers treat the initial development set in SMT as the source domain and the

test set as the target domain in DA. Reference [9] trained an SMT baseline system using out-of-domain corpora and then used in-domain resource to improve the performance in-domain. In their method the development set is composed of in-domain data and out-of-domain data. Reference [10] proposed an alignment-based discriminative framework for string similarity. Reference [11] presented an algorithm which takes account of semantic information and word order information implied in the sentence to calculate the similarity between those sentences with very short sentence length. Reference [12] selected sentences for development set according to the phrase weight estimated from the test set.

As mentioned above, many methods have their limitations for most of the methods are based on the test data to choose the training data and development set. But for a practical system, the test data is unknown. Moreover, it would cost too much time and effort to choose the training data and development set for each test data. Therefore, we would like to focus on approaches to selecting both the training data and development set without guidance of the test set.

3. Our Approaches to Selecting Training Data

Based on the analysis above, we know that to process the training data for an SMT system, there are two key issues: ① how to filter the noise from training corpus to improve the quality of training data? ② how to fix the size of training data to trade off the coverage of corpus and computation load so that the system may have better performance given an acceptable speed.

To deal with these two issues, we propose an integrated approach which includes two steps: ① filter the noise in the training data to get the optimized corpus. ② a sentence weight is estimated by the weights of all basic units in the sentence, and the more informative sentences are selected from the optimized corpus to build a compact training set. The compact training set is used to train the translation model. The ideas of our approach is shown as Figure 1.

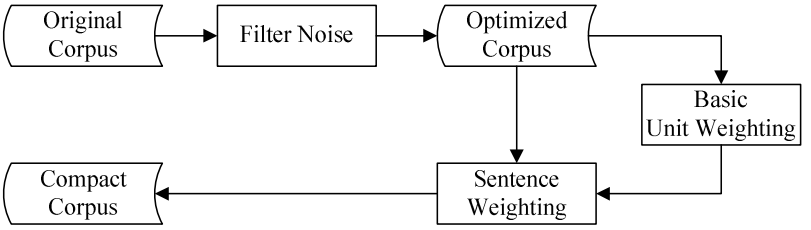

Figure 1. The framework of train data processing.

3.1. Noise filtering

In order to filter the noise in the training data, two simple strategies are proposed. One is based on the length ratio (*LR*) and the other is based on the translation ratio (*TR*).

(1) *The Strategy based on LR*: In our opinion, for a parallel sentence pair, the length ratio of two sentences in a sentence pair should be distributed in a certain range. Otherwise, if their length ratio is out of a range, we think the two sentences in this pair are not properly corresponding to each other. We should filter such sentence pair as noise. We define the length ratio of two sentences in a sentence pair as follows:

$$LR = \frac{|s|_{tgt}}{|s|_{src}} = \frac{l}{m} \tag{1}$$

where, $|s|_{src} = m$ is the length of the source language sentence, and $|s|_{tgt} = l$ is the length of the target language sentence. In our approach, if the *LR* value of two sentences in a sentence pair is bigger than the given threshold value, the sentence pair will be filtered out.

(2) *The strategy based on TR*: Intuitively, if two sentences in a sentence pair are closely corresponding to each other, their words in two sentences should have good correspondence. That means the translation of words in the source language sentence should have a greater chance to appear in the target language sentence. Therefore, we use the translation ratio as a measure to filter the noise. If the *TR* value for two sentences in a sentence pair is less than the given threshold value, the sentence pair will be filtered out as noise. We use a bilingual dictionary to estimate how many word pairs co-occurred in two sentences. We calculate the translation ratio using the following formula:

$$TR = \frac{\sum \#(word - pair)}{|s|_{src}} \tag{2}$$

where, $|s|_{src}$ is the length of source language sentence, $\sum \#(word - pair)$ denotes the total number of words in the source sentence whose translations appear in the target language sentence. According to the distribution of translation ratio on a large scale corpus, we can fix the threshold value for *TR* to filter the noise.

We can filter the training corpus either by using the strategy based on *LR* or using the strategy based on *TR*, or by combing the two strategies on *LR* and *TR* gradually: first filter the noise sentence pairs by *LR* score and then further refine the training corpus by using *TR* measure score.

3.2. Representative data selection

In order to reduce the size of the training data but keep the coverage, we have to select the representative sentences which can cover more information of the entire original corpus. According to the information theory, the information contained in a statement can be measured by the negative logarithm of the probability of the statement [13, 14]. Therefore, we use this method to estimate the weight of a sentence and select the sentences according to their weights.

Take the phrase-based translation model (PBTM) as an example. In PBTM, the phrase is considered as the basic translation unit [15] and the translation is performed based on the phrase translation probability. We think it is reasonable to estimate the weight of a sentence pair according to the information contained in their phrases. In order to estimate the weight of each phrase and then estimate the weight of a sentence pair based on the weights, we consider the following two factors: ① the information contained in a phrase, and ② the phrase length. Here the phrase can be considered as source phrase or target phrase.

As we mentioned above, the information contained in a phrase can be calculated by formula (3) below:

$$I(f) = -\log_2 p(f) \tag{3}$$

where f is a phrase, and $p(f)$ is the translation probability of phrase f.

Generally speaking, in PBTM the longer phrases can lead to better performance. So we take the length of phrases into account to construct the weight. We assign the weight to each phrase by using formula (4):

$$w(f) = \sqrt{|f|} \cdot I(f) \tag{4}$$

where $|f|$ is the length of the phrase. The reason to use the square root of the length instead of the length itself is just for data smoothing. In order to cover more phrases in the chosen corpus, we assign higher weight to the sentence pair which has more unseen phrases.

After having the phrase weights by using formula (4), we design two methods to estimate the weight for a sentence pair. The first one is defined by the following formula (5):

$$W_1(s) = \frac{\sum_i w(f_i)E(f_i)}{|s|} \tag{5}$$

where s is a sentence, and its length is $|s|$. $E(f_i)$ is defined as formula (6):

$$E(f_i) = \begin{cases} 0, & \text{if } f_i \text{ is contained in the new training corpus} \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where f_i is a phrase contained in the sentence s of a chosen sentence pair, but not contained in the new choosing corpus. We call such f_i as the unseen phrase. If a phrase also occurs in a sentence of the new choosing corpus, the weight is set to zero. If we only consider the new phrases, the longer sentence will tend to get higher score because it contains more unseen phrases. So we punish the sentence weight by the sentence length.

Another method to estimate the sentence weight is shown as the following formula (7):

$$W_2(s) = \begin{cases} \dfrac{\sum_i w(f_i)E(f_i)}{\sum_i E(f_i)}, & \sum_i E(f_i) \neq 0 \\ 0, & \sum_i E(f_i) = 0 \end{cases}. \tag{7}$$

The variables of formula (7) are defined as the same as in formula (6).

The difference between the two methods lies that the first one ($W_1(s)$) uses the sentence length as denominator but the second one ($W_2(s)$) uses the sum number of unseen phrases as the denominator. This may cause that $W_1(s)$ tends to select those sentences with more unseen phrases and $W_2(s)$ tends to select those sentences with less-probability phrases.

We select the new training data by using the algorithm shown in Figure 2. In the algorithm *OriTrain* denotes the original training data and *NewTrain* means the new training data.

```
1: Initialization: NewTrain = null
2: While (OriTrain != null):
3: Calculate the sentence weight;
4: Sort;
5: Add the sentence with highest weight in the NewTrain;
6: Delete the sentence with highest weight in the OriTrain;
```

Figure 2. The algorithm to select new training data.

4. Our Approaches to Selecting Development Data

As we know the development set is used to tune the translation parameters which have great influence on the translation quality and system robustness. In order to get the optimized parameters, MERT is usually employed on the development set

to tune the parameters. However, the method based on MERT often consumes too much time and too much computation resources until it converges, especially when the development set is in a large scale. Even though, we cannot be sure if the parameters are optimized or not. In most cases, the BLEU scores are very different for the same test set if the parameters are tuned with a different development set. Therefore, in order to consider the whole performance of a system including the speed of MERT, the translation quality and the robustness, we focus on the following two problems: ① what scale of the development set is adequate and what kind of development sentence is more contributable to tune the parameters that can achieve an optimal and robust performance? ② how to choose the appropriate sentences from the development data to form the new development set?

Intuitively speaking, if the development set is more similar to the test set, the better translation performance may be achieved on the parameters tuned on the development set. Unfortunately, in most cases the test data is unknown. So the general similarity measure is not practical for our task and we can only choose the development set by relying on its own information.

Since the development set is often much smaller than the training corpus, we can extract more effective features. An intuitive idea is if the extracted sentences can cover more information (such as word, phrase and structure) of the original development data, the new development set is better. So we select such sentences which can cover more information of the entire original development data. Because the word is a special phrase, we mainly focus on the phrase-coverage and structure-coverage to introduce our methods.

4.1. Phrase-coverage-based method

As described in training data selection, the phrase is an important feature for PBTM. So we take the phrase coverage as the metric and call this method the phrase-coverage-based method (PCBM).

We take two aspects into account to estimate the weight of phrase: the information it contains and the length of the phrase. Here the phrase can be a source phrase or a target phrase. We first extract all the phrases in the sentences of the development set and limit the maximum length of the phrases as 4 words in order to avoid the problem of data sparseness. Then we classify the phrases into different classes according to their length, and finally we calculate the probabilities respectively. For example, for the phrase f with length $|f| = n$, the phrase probability in the development set can be calculated by formula (8):

$$p(f) = \frac{count(f)}{\sum_{|f_i|=n} count(f)} \tag{8}$$

where $count(f)$ is the occurrence number of f in the development set and $\sum_{|f_i|=n} count(f_i)$ is the sum number of all phrases with the same length n. According to the information theory, the information carried by phrase f is:

$$I(f) = -\log_2 p(f). \tag{9}$$

We use formula (10) to calculate the weight for each phrase.

$$w(f) = \sqrt{n} \cdot I(f). \tag{10}$$

Then, we use formula (11) to calculate the weight for each sentence:

$$W_{PH}(s) = \frac{\sum_{f \in F} w(f)}{|s|}. \tag{11}$$

The definition of the phrase weight is similar as it is in training data selection, see formula (4). The only important difference is: in estimating the sentence weight for development set, all phrases are considered, not just the unseen phrases. The new development data is selected according to the weights of the sentences. The sentences with higher weights have priority to be selected.

4.2. Structure-coverage-based method

As we have seen that PCBM only considers phrases, a surface feature, to estimate the weight of sentences. It doesn't contain any deep features, such as sentence structure. So we propose another method to use sentence structure features to help choosing the development set. We name this method as structure-coverage-based method (SCBM).

In this method, our purpose is to extract all sentences which can cover the majority structures of the development set. We first parse all source sentences in the entire original development set into phrase-structure trees with the Stanford parser [16]. Then we analyze the subtrees contained in the phrase-structure trees of source sentences, and finally extract all sentences which can cover more subtrees. In order to avoid the problem of data sparseness, we use the subtrees whose depth are between two and four levels. An example of the subtree is shown in Figure 3.

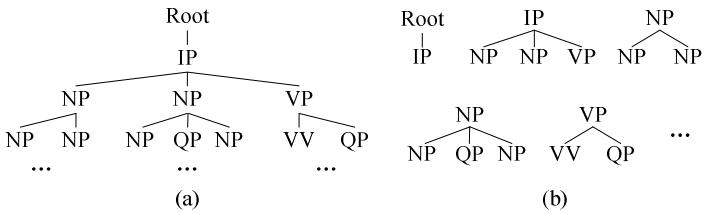

Figure 3. (a) Phrase-structure tree; (b) Subtrees (depth = 2).

We consider two factors to estimate the weight of the subtrees: depth and information. For a subtree t, its depth is $|t|$. Assume its probability $p(t)$, which can be estimated from the development set. The information contained in the subtree t is calculated by formula (12).

$$I(t) = -\log_2 p(t) \tag{12}$$

Then the weight of each subtree in the development set is calculated by formula (13):

$$w(t) = \sqrt{d} \cdot I(t) \tag{13}$$

where d is the depth of the subtree.

The score of a source sentence is calculated by the following formula (14):

$$S_{SCBM}(s) = \frac{\sum_{t \in T} w(t)}{|s|} \tag{14}$$

where T is the set of subtrees contained in sentence s. Those sentence pairs will have priority to be selected if the scores of their source sentences are bigger than the given threshold.

5. Experiments

5.1. Results on training corpus selection

On training data processing, we have done experiments on CWMT'2008 (China Workshop on Machine Translation, 2008) corpus. We randomly chose 20 million words as the original training corpus to construct our experiments on Chinese-to-English translation task. And we randomly selected 400 sentences from the development set as the test set. All our experiments have been carried out on the free toolkit Moses[a] and all the parameters are set as their defaults. The translation results are evaluated by BLEU metrics [17].

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

5.1.1. *Experiments on filtering noise method*

We adopt three methods to filter the noise sentences: the first one is based on *LR* approach, the second one is based on *TR* approach, and the third one is based on the combination of *LR* and *TR*. The combination approach is called *CR*.

- Using *LR* approach

We have done a survey on length ratio of the English sentence length to Chinese sentence length on a bilingual Chinese-English training data, which includes 2,840,000 sentence pairs. The statistical results are shown as Figure 4. Where the vertical axis denotes the number of the bilingual sentences and the horizontal axis denotes the length ratio of the English sentence length to Chinese sentence length.

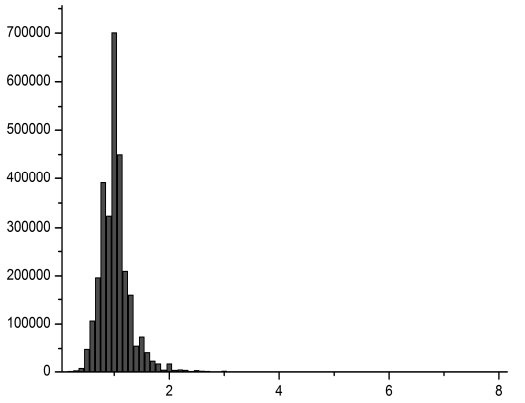

Figure 4. Ratio distribution of English sentence length to Chinese sentence length.

Here, all the Chinese sentences are segmented with the segmentation tool Urheen [18] and their length is calculated based on the word numbers. In the statistical results, the length ratios are distributed within a certain range 0.1~8.1 and more than 96% ones are between 0.6 and 1.7. So we take these two values as thresholds. That means if the length ratio of two sentences in a sentence pair is out of the range from 0.6 to 1.7, the sentence pair will be discarded.

- Using *TR* approach

In this approach, we first obtain the lemma of each word in the target language by using morph toolkit[b]. We use a English-to-Chinese dictionary which contains more than 950 thousand entries to calculate the translation ratio of each sentence

[b] http://www.informatics.susx.ac.uk/research/groups/carroll/morph.html

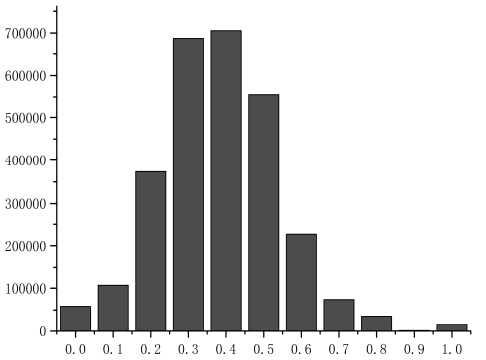

Figure 5. The translation ratio of bilingual sentence pairs.

pair. The statistical results are shown as Figure 5, where the vertical axis denotes the number of the bilingual sentences and the horizontal axis denotes the translation ratio of the English sentences to the Chinese sentences.

As it is shown in Figure 5, in the training corpus, the translation ratio of about 95% sentence pairs is higher than 0.2. So we take this value as the threshold. That means the sentence pairs whose translation ratio is less than 0.2 are regarded as noise data.

- Using ***CR*** approach

We first use *LR* to filter the bilingual sentences to get the new training data and then use *TR* to filter the new training data to get the final training data.

- System performance

We compared the system performance on different training data that are filtered by using *LR*, *TR*, and *CR* approach respectively. The results are shown in Table 1.

Table 1. Results of noise filtering methods for the training corpus.

	Original	**LR**	**TR**	**CR**
Number of Words (M)	20.00	19.67	19.72	19.40
Number of Sentences	778,079	757,092	762,108	741,974
BLEU Score	0.2132	0.2128	**0.2153**	0.2135

In Table 1, '*Original*' means the results using all original training data. '*LR*' means the results on the corpus filtered by the *LR* method. About 0.33 million words are filtered out and the BLEU score has 0.04% decline. In the forth column '*TR*' means the result on the corpus filtered by the *TR* method. About 0.28 million words are filtered out and the BLEU score has been improved for

0.21%. In the last column '*CR*' means the results of corpus filtered by combining two methods *LR* and *TR*. The BLEU score is almost the same as it is on the original corpus although about 0.6 million words have been filtered out. From the Table 1, it is clear that the *TR* method is more robust and effective. This is because *TR* method makes use of bilingual dictionary information, and the precision is higher than the results of the *LR* method. However, the *LR* method still gets a competitive performance compared to using all the training data although it never uses any additional resource.

5.1.2. *Experiments on training data selection methods*

In experiments on training data selection, we selected different training corpus with different size. We have tested the following four methods: ① selecting the sentences randomly from the training data to combine the baseline system; ② weighing the sentences only considering the quantity of the unseen source phrases without considering the weight of the source phrases. We call this method as *unWP*; ③ considering both the quantity and weight of the unseen source phrases by the constraints of formula (5). This method is called *WP*1; and ④ consider both the quantity and weight of the unseen source phrases by the constraints of formula (7). This method is called *WP*2. All the experiments have been done using the four methods. Table 2 gives the BLEU scores of different methods, Table 3 gives the word recall, and Table 4 gives the sentence percentage.

Table 2. The BLEU scores under various extraction methods on training data.

Number of Words (M)	**Baseline**	***unWP***	***WP*1**	***WP*2**
2	0.1357	0.1614	0.1726	0.1673
4	0.1384	0.1842	0.1918	0.1863
6	0.1468	0.1887	0.1955	0.1896
8	0.1511	0.1947	0.2010	0.2026
10	**0.1532**	**0.2033**	**0.2060**	**0.2114**
12	0.1609	0.2059	0.2071	0.2171
14	0.1724	0.2055	0.2098	0.2124
16	0.1990	0.2100	0.2118	0.2124
18	0.2095	0.2046	0.2121	0.2127
20	**0.2132**	0.2132	0.2132	0.2132

From Table 2 we can see that our extraction methods achieve the better performance compared to the baseline system with the same training data. Moreover, the system performance under the *WP*2 method, which only uses half of all the words, is comparable with the baseline system with all the original training data: 0.2114 *vs* 0.2132.

Table 3. The word recall under various extraction methods.

Number of Words (M)	**Baseline**	***unWP***	***WP*1**	***WP*2**
2	24.8%	55.7%	67.0%	63.5%
4	30.6%	74.9%	78.7%	75.1%
6	36.8%	83.1%	85.1%	81.0%
8	42.6%	88.3%	89.2%	85.4%
10	**45.9%**	**91.8%**	**92.3%**	**88.7%**
12	51.1%	94.4%	94.6%	91.5%
14	60.4%	96.3%	96.4%	94.2%
16	82.8%	97.9%	97.9%	96.7%
18	95.7%	99.2%	99.2%	98.8%
20	100.0%	100.0%	100.0%	100.0%

Table 4. The sentence percentage under various extraction methods.

Number of Words (M)	**Baseline**	***unWP***	***WP*1**	***WP*2**
2	9.3%	7.5%	8.2%	11.4%
4	18.7%	16.5%	17.2%	21.7%
6	27.9%	25.7%	26.5%	31.5%
8	37.1%	35.3%	36.0%	41.1%
10	**46.5%**	**45.2%**	**45.8%**	**50.7%**
12	55.7%	55.2%	55.8%	60.2%
14	65.3%	65.5%	66.0%	69.7%
16	77.7%	76.3%	76.7%	79.3%
18	89.1%	87.3%	87.6%	88.8%
20	100.0%	100.0%	100.0%	100.0%

From the results of Table 2, Table 3, and Table 4, we can easily see that the data selected by our methods can cover more phrases and get a higher score only using smaller size data. Our method can get the competitive performance with much less data, and this largely reduces the computational load. For example, when we use only half of the training words (10M), the baseline only covers 45.9% words, while *unWP* method can cover 91.8% words, *WP*1 method can cover 92.3% words, and *WP*2 method covers 88.7% words. All the word coverage obtained by our methods are much higher than the baseline. As we mentioned in Sub-Section 3.2, since *WP*1 prefers unseen phrases more but *WP*2 prefers higher weighted phrases more, so the *WP*1 recall is higher than *WP*2 but *WP*2 performance is higher than *WP*1. The reason is that *WP*2 can achieve more accurate probabilities with more compact training data. The BLEU score of *WP*2 method in our experiment is 0.2114, 5.82% higher than the baseline 0.1532, and it is only 0.18% lower than the system that uses all the available data (its score is 0.2132). However, under the same number of words (10M), the different training data obtained by our methods have almost the same quantity of sentences. The sentence percentages are 46.5%, 45.2%, 45.8%, and 50.7%, respectively.

In summary, with the same size of data, our methods can extract more informative sentences and cover more words. Moreover, we use only half of the data to get a competitive performance compared to the baseline using all the data. The system on training data selected by using *unWP* and *WP* (*WP*1 and *WP*2) methods always outperforms the baseline, especially when the training data is small.

5.2. Results on development corpus selection

We have also done experiments on development data selection on CWMT'2009[c] and IWSLT'2009[d] translation tasks, both in bidirectional Chinese-English translation. The former is in news domain in formal text, and the latter is in travel domain in spoken language. For CWMT'2009 task, we randomly selected 400 sentences from the development set as the test set, and took the left as the development set. For IWSLT'2009 tasks, we employed BTEC corpus on Chinese-to-English task and Challenge English-to-Chinese task. Table 5 shows the information of the corpora.

[c]http://www.icip.org.cn/cwmt2009
[d]http://mastarpj.nict.go.jp/IWSLT2009/

Table 5. Development data for data selection.

	Task	Development Set		Test Set
		Sentence	Words	Sentence
CWMT'2009	C-E	2,876	57,010	400
	E-C	3,081	55,815	400
IWSLT'2009	C-E	2,508	17,940	469
	E-C	1,465	12,210	393

On each task, we select sentences randomly to build the baseline. Then we select the different scale of development data for MERT using four approaches we proposed: ① only consider the Chinese phrases (*Ch*) under PCBM method; ② only consider the English phrases (*En*) under PCBM method; ③ consider both the Chinese and English phrases (*Ch* + *En*) simultaneously under PCBM method; and ④ use structure-based method under SCBM method. For SCBM method, we only use the Chinese sentences and parse them using Stanford parser. The results are shown in Table 6 and Figure 6, where '*Recall*' means the word recall of the new development set and '*Percentage*' denotes the ratio of the new development set to the original development data.

Table 6(a). The translation results of the C-E task in CWMT'2009.

Number of Words (10K)		1.0	2.0	3.0	4.0	5.0	5.6
Bas.	BLEU	0.1157	0.1367	0.1506	0.1615	0.1668	0.1709
	Recall	35.51%	56.63%	69.45%	77.10%	90.10%	100%
	Perc.	13.32%	35.05%	58.38%	75.17%	91.93%	100%
Ch	BLEU	0.1258	0.1567	0.1605	0.1678	0.1724	0.1709
	Recall	51.28%	73.39%	86.84%	95.49%	99.67%	100%
	Perc.	9.39%	21.45%	35.92%	55.32%	84.21%	100%
En	BLEU	0.1239	0.1517	0.1555	0.1703	0.1688	0.1709
	Recall	47.23%	69.87%	83.67%	93.39%	99.38%	100%
	Perc.	10.22%	22.39%	36.96%	56.61%	84.11%	100%
Ch + En	BLEU	0.1240	0.1491	0.1569	0.1690	0.1714	0.1709
	Recall	50.09%	72.86%	86.18%	94.70%	99.70%	100%
	Perc.	9.35%	21.45%	35.78%	54.03%	83.76%	100%
Stru.	BLEU	0.1215	0.1488	0.1600	0.1610	0.1692	0.1709
	Recall	45.91%	68.33%	82.16%	92.51%	98.63%	100%
	Perc.	9.91%	23.65%	36.97%	57.08%	85.22%	100%

Table 6(b). The translation results of the E-C task in CWMT'2009.

Number of Words (10K)		**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**5.6**
Bas.	BLEU	0.0666	0.0801	0.0856	0.0854	0.0968	0.0955
	Recall	34.12%	50.32%	65.30%	79.67%	92.66%	100%
	Perc.	12.76%	36.09%	59.07%	75.53%	92.76%	100%
Ch	BLEU	0.0738	0.0898	0.0926	0.0994	0.1004	0.0955
	Recall	49.14%	71.06%	84.43%	93.59%	99.03%	100%
	Perc.	10.39%	22.30%	35.77%	52.84%	78.55%	100%
En	BLEU	0.0667	0.0854	0.0942	0.0960	0.1000	0.0955
	Recall	45.10%	67.57%	82.24%	92.14%	98.36%	100%
	Perc.	9.80%	21.65%	34.66%	51.67%	76.99%	100%
Ch + En	BLEU	0.0721	0.0811	0.0967	0.1014	0.0990	0.0955
	Recall	48.11%	70.34%	83.98%	93.11%	98.73%	100%
	Perc.	9.80%	21.55%	34.60%	51.31%	76.92%	100%
Stru.	BLEU	0.065	0.0821	0.0911	0.0952	0.0987	0.0955
	Recall	46.27%	66.35%	80.59%	91.33%	97.56%	100%
	Perc.	11.28%	23.41%	36.02%	52.77%	77.13%	100%

Table 6(c). The translation results of the C-E task in IWSLT'2009.

Number of words (10K)		**3.0**	**6.0**	**9.0**	**12.0**	**15.0**	**17.9**
Bas.	BLEU	0.1542	0.1862	0.1938	0.2133	0.2283	0.2324
	Recall	33.09%	52.32%	68.01%	80.22%	91.84%	100%
	Perc.	16.87%	33.53%	49.44%	65.71%	82.89%	100%
Ch	BLEU	0.1803	0.1990	0.2088	0.2248	0.2376	0.2324
	Recall	49.70%	72.70%	85.42%	94.38%	98.90%	100%
	Perc.	9.09%	22.49%	38.44%	56.62%	76.28%	100%
En	BLEU	0.1532	0.2037	0.2155	0.2241	0.2318	0.2324
	Recall	46.24%	68.93%	82.76%	92.90%	97.84%	100%
	Perc.	10.37%	24.44%	40.75%	58.61%	77.71%	100%
Ch + En	BLEU	0.1632	0.1785	0.2137	0.2309	0.2311	0.2324
	Recall	48.35%	71.64%	85.16%	94.93%	99.45%	100%
	Perc.	9.41%	23.21%	38.80%	57.18%	76.67%	100%
Stru.	BLEU	0.1566	0.1810	0.2011	0.2012	0.2319	0.2324
	Recall	45.72%	70.25%	80.26%	91.36%	95.22%	100%
	Perc.	9.85%	22.49%	39.12%	59.42%	78.36%	100%

Table 6(d). The translation results of the E-C task in IWSLT'2009.

Number of words (10K)		2.0	4.0	6.0	8.0	10.0	12.0
Bas.	BLEU	0.2787	0.2961	0.2709	0.2932	**0.3051**	0.3080
	Recall	26.59%	41.30%	50.75%	55.85%	**77.01%**	97.83%
	Perc.	9.35%	24.30%	43.41%	62.32%	80.55%	98.29%
Ch	BLEU	0.2657	0.3062	0.3105	0.3182	0.3212	0.3089
	Recall	56.77%	79.10%	93.06%	98.16%	100.00%	100.00%
	Perc.	9.76%	23.82%	41.16%	58.91%	78.02%	97.06%
En	BLEU	0.2753	0.2933	0.3075	0.3116	0.3126	0.3107
	Recall	51.67%	76.34%	90.13%	96.82%	99.00%	100.00%
	Perc.	10.03%	24.30%	41.30%	59.04%	77.88%	97.68%
Ch + En	BLEU	0.2712	0.2928	0.3133	0.3201	**0.3202**	0.3055
	Recall	55.18%	80.18%	93.23%	98.41%	**99.92%**	100.00%
	Perc.	9.76%	24.37%	40.55%	58.98%	77.47%	97.54%
Stru.	BLEU	0.2680	0.2961	0.2947	0.3059	0.3117	0.3070
	Recall	54.69%	75.40%	91.62%	97.28%	98.86%	100.00%
	Perc.	10.03%	24.12%	42.76%	60.22%	78.45%	97.68%

From the experimental results shown in Table 6 and Figure 6, we can clearly see that when compared to the baseline system, the development corpus selected by our methods can get higher performance with the same quantity of data. When the development corpus is in large scale, our method can select more informative sentences for MERT. For the phrase-coverage-based method, when we consider both the Chinese phrases and the English phrases, the performance is better and more robust compared to the methods which only consider monolingual phrases. This is because the sentences extracted by using this method can cover the information both in source language and target language, and make the translation parameters more robust.

Where the horizontal axis is the scale of the development corpus, the unit is one thousand words. The vertical axis is BLEU score of the test set using the parameters trained on the corresponding development data.

A notable phenomenon is that we can get even higher score using a part of the development data than using all data. For example, in Figure 6(d), when we use 10 thousand words for MERT, the performance is better than using 12 thousand words. We present the recall of words for the baseline method and PCBM method which considers bilingual phrases in Table 6(d). In this table, the baseline's recall is only 77.01% while the PCBM's recall is 99.92% ($Ch + En$)

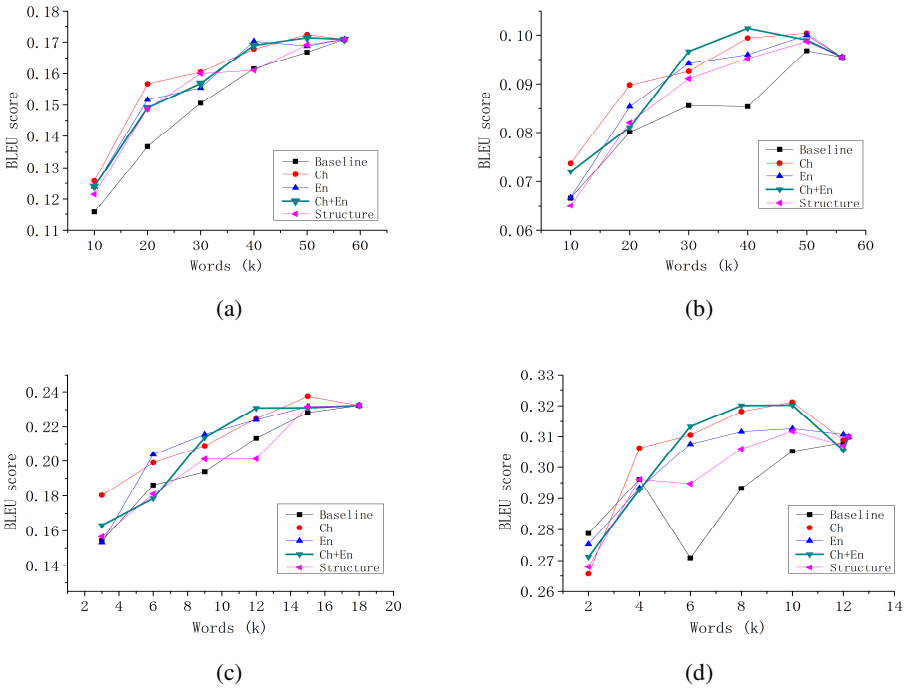

Figure 6. Results of data selection for development data: (a) CWMT09 C-E; (b) CWMT09 E-C; (c) IWSLT09 C-E; (d) IWSLT09 E-C.

when the development data has 10 thousand words; almost all the words have been covered. That means adding more data to development set brings little improvement to the recall of words, but imports many redundant sentences and undermines the performance of the translations.

The structure-coverage-based method performs not as well as the phrase-coverage-based method does, though it is better than the baseline. This is probably due to the following reasons: ① The precision of the parser is not good enough and it often imports some errors into the parsing results and decreases the performance of the translation system; ② The problem of data sparseness is very serious when we adopt the subtrees as the feature to select sentences; ③ The translation model we used is a phrase-based model, so it can exert more merit of phrases but not subtrees; and ④ Both sides of Chinese and English phrases are considered in the PCBM, which can achieve more accurate parameters on such development set. For these reasons, it is not necessary to try the combination of phrase-coverage-based method and the structure-coverage-based method.

6. Conclusions

In this paper, we propose series of approaches to improving the quality of the training corpus and the development corpus for SMT system development. For training data selection, we propose the methods to filter the noise parallel sentence pairs based on the length ratio and the translation ratio strategies. The experiments have shown that we can select more informative sentences to build a more compact training corpus using the weighted-phrase method, which can achieve a competitive performance compared to the baseline system using all training data. This can greatly decrease the computing resources and improve the translation speed.

For development data selection, we propose two methods respectively based on phrase-coverage and structure-coverage. Our experiments have shown that both methods outperform the baseline system. When we consider the bilingual phrases, the performance is better and more robust. If the word recall is bigger than 95%, it will achieve the best performance.

The investigation on the approaches to selecting training data and development data is very meaningful for SMT system development. The proposed approaches are very helpful to developing a better SMT system based on the limited training data or in the limitation of computing resources.

However, there is so much work left for further research. The next step is to verify our methods to more effectively improve the performance of statistical machine translation systems. We will do more additional experiments in various cases, including on the same data, in different domains, on different topics, or other cases.

Acknowledgements

The research work described in this paper has been funded by the Natural Science Foundation of China under Grant No. 60975053 and 61003160, and supported by the External Cooperation Program of the Chinese Academy of Sciences as well. This research is also partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

[1] Y. Zhou, Y. He and C. Zong, The CASIA Phrase-Based Statistical Machine Translation System for IWSLT 2007, *Proc. of IWSLT'2007*, Trento, Italy, 2007.

[2] C. Chai, The Design and Implementation of Decoder for Statistical Machine Translation, Masters thesis (in Chinese), CASIA, 2007.
[3] P. Resnik and N. A. Smith, Articles The Web as a Parallel Corpus, *Computational Linguistics*, Vol. 29, No. 3, 2003, pp. 349–380.
[4] M. Snover, B. Dorr and R. Schwartz, Language and Translation Model Adaptation using Comparable Corpora, *Proc. of EMNLP*, 2008, pp. 857–866.
[5] M. Eck, S. Vogel and A. Waibel, Low Cost Portability for Statistical Machine Translation based on N-gram Coverage, *Proc. of the 10th MT Summit*, Phuket, Thailand, 2005, pp. 227–234.
[6] Y. Lü, J. Huang and Q. Liu, Improving Statistical Machine Translation Performance by Training Data Selection and Optimization, *Proc. of EMNLP-CoNLL*, Prague, 2007, pp. 343–350.
[7] K. Yasuda, R. Zhang, H. Yamamoto and E. Sumita, Method of Selecting Training Data to Build a Compact and Efficient Translation Model, *Proc. of the 3rd IJCNLP*, India, 2008, pp. 655–660.
[8] S. Matsoukas, A.-V. I. Rosti and B. Zhang, Discriminative Corpus Weight Estimation for Machine Translation, *Proc. of EMNLP*, Singapore, 2009, pp. 708–717.
[9] H. Wu, Haifeng Wang and Chengqing Zong, Domain Adaptation for Statistical Machine Translation with Domain Dictionary and Monolingual Corpora, *Proc. of the 22nd COLING*, 2008, pp. 993–100.
[10] S. Bergsma and Grzegorz Kondrak, Alignment-Based Discriminative String Similarity, *Proc. of the 45th ACL*, 2007, pp. 656–663.
[11] Y. Li, David Mclean, Zuhair A. Bandar, James D. O'shea and Keeley Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, *IEEE Transactions on Knowledge and Data Engineering*, 18(8), 2006, 1138–1150.
[12] P. Liu, Y. Zhou and C. Zong, Approach to Selecting Best Development Set for Phrase-based Statistical Machine Translation, *Proc. of the 23rd PACLIC*, Hongkong, 2009, pp. 325–334.
[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
[14] D. Lin, An Information-Theoretic Definition of Similarity, *Proc. of the 5th ICML*, 1998, pp. 296–304.
[15] P. Koehn, F. J. Och and D. Marcu, Statistical Phrase-Based Translation, *Proc. of HLT-NAACL*, Edmonton, 2003, pp. 48–54.
[16] R. Levy and C. D. Manning, Is it Harder to Parse Chinese, or the Chinese Treebank?, *Proc. of the 41st ACL*, Sapporo, Japan, 2003, pp. 439–446.

[17] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, Bleu: A Method for Automatic Evaluation of Machine Translation, *Proc. of the 40th ACL*, 2002, pp. 311–318.
[18] K. Wang, Chengqing Zong and Keh-Yih Su, A Character-Based Joint Model for Chinese Word Segmentation, *Proc. of the 23rd COLING*, Beijing, China, August 23–27, 2010, pp. 1173–1181.
[19] C. Zong and Mark Seligman, Toward Practical Spoken Language Translation, *Machine Translation*, 19(2), 2005, 113–137.