Nanjing University's System Report for NIST MT09 Workshop

Shujian HUANG, Yinggong ZHAO, Boyuan LI,
Qiufeng WU, Xinyu DAI, Jiajun CHEN

Natural Language Processing Research Group
Department of Computer Science and Technology, Nanjing University
{huangsj, zhaoyg, liby, wuqf, daixy, chenjj}@nlp.nju.edu.cn

Abstract. This paper describes our participation (NJU-NLP) in the Chinese-to-English Progress Test of the NIST Open MT09 evaluation. We built a phrase-based machine translation system with the help of MOSES and tried several methods to improve the result. Our efforts include pre-segmenting long train sentence pairs into shorter ones, phrase table smoothing, phrase table filtering. Details of these techniques as well as our evaluation results are reported.

1 Introduction

This paper describes the system of NJU-NLP (Natural Language Processing Research Group, Nanjing University) in the NIST Open MT09 evaluation. We participate in the task of Chinese-to-English Progress Test. As a first time participant, we employ the open source phrase-based system MOSES [1] as our main system. Besides, we have tried the following techniques to improve the translation result.

Firstly, we split the long sentence pairs in the training corpus into shorter ones, in order to get better alignment quality [2, 3]. Secondly, we smooth the phrase table [4, 5] to get better phrase scores. Thirdly, we filter the phrase table according to the word coverage of development and test data right after the phrase are extracted to reduce the size of the table. We describe these techniques in detail in the following sections.

2 Sentence Segmentation

[1] In the phrase-based statistical machine translation model [1], the knowledge on phrase translation and phrase reordering is learned from the bilingual corpora. However, words may be poorly aligned in long sentence pairs in practice, which will then do harm to the following steps of the translation. On the other hands, training a system using long sentence pairs usually cost much more memory and

[1] We have improved our work on sentence segmentation after the evaluation and reported it to IALP2009[3].

CPU time, which is much less efficient. In order to make good use of the information carried by long sentence pairs, it's necessary to segment long sentences into shorter ones.

2.1 Definition

For a given sentence pair (f, e), a *Segmentation* s is defined as a sequence of sub-sentence pairs $(f_1, e_1), (f_2, e_2), ..., (f_n, e_n)$, which satisfies Formula 1,

$$\begin{aligned} \overline{f_{a_1} f_{a_2} f_{a_3} \cdots f_{a_{n-1}} f_{a_n}} &= f \\ \overline{e_{a_1} e_{a_2} e_{a_3} \cdots e_{a_{n-1}} e_{a_n}} &= e \end{aligned} \tag{1}$$

where f_i and e_j are permutations on $[1, n]$. We define *Sentence Segmentation* as the problem of searching for the best *Segmentation* s of given sentence pair (f, e) under some probability model (see Formula 2), where S is the set of all possible *Segmentations*.

$$\begin{aligned} s &= argmax_{s \in S} Pr(s) \\ &= argmax_{s \in S} Pr((f_1, e_1), (f_2, e_2), \dots, (f_n, e_n)) \end{aligned} \tag{2}$$

In this paper, we extend Xu et al. [2] and use the following probability model which is based on IBM Model1 (see Formula 3 and 4), where a is an alignment on (f_i, e_i) and A is the set of all possible alignments.

$$\begin{aligned} Pr((f_1, e_1), (f_2, e_2), \dots, (f_n, e_n)) &= Pr(f_1, e_1) Pr(f_2, e_2) \dots Pr(f_n, e_n) \\ &= \mathit{\Pi}_{i=1}^{n} Pr(f_i, e_i) \end{aligned} \tag{3}$$

$$Pr(f_i, e_i) = \frac{\epsilon}{(l+1)^m} \mathit{\Sigma}_{a \in A} \mathit{\Pi}_{j=1}^{m} t(f_j | e_{a_j}) \tag{4}$$

2.2 Search Strategy

Obviously, it is not feasible to enumerate all possible *Segmentations* to find the best one. To solve the problem, we use a greedy-based search algorithm which just looks for the best 2-part segmentation (Formula 5), and recursively segments the sentence.

$$s = argmax_{s \in S} Pr(f_1, e_1) Pr(f_2, e_2) \tag{5}$$

For the stop criterion of the search, we use a predefined length threshold k. If the length of the segmented sentences is still larger than k, segmentation will be performed on the sub sentences recursively.

3 Phrase Table Smoothing

It is well known that phrase table training faces the problem of sparseness. There are a lot of phrases that occur just one or two times. To better estimate the probabilities of those rare phrases and make the whole distribution more robust, we perform a smoothing on the extracted phrase table.

The basic idea here is to mix the original distribution, which often has a high complexity and a high variance, with certain distribution with lower complexity and lower variance. Chen and Goodman [4] give a empirical study of smoothing techniques for language model. Foster et al. [5] report good results on phrase table smoothing. In the evaluation, we employ an Absolute Discounting method for the smoothing task (Formula 6).

$$P_{abs}(s|t) = \frac{\max\{C(s,t) - D, 0\}}{C(t)} + \frac{D * N_{1+}(t)}{C(t)} p(s) \tag{6}$$

$$D = \frac{n_1}{n_1 + 2n_2}$$

where s, t are source and target part of the phrase, $C(s,t)$ is the number of occurrences of s and t as a phrase pair, n_i is the number of phrases that occur i times. N_{1+} is the number of phrases that occur more than one time.

4 Phrase Table Filtering

The whole training data provided by NIST has more than 5 million sentence pairs, which is quite large to process. We filter the phrase table right after all the phrases are extracted. Table 1 shows the filtering algorithm, where C refers to the given development and test data, and P refers to the extracted phrase list. The phrase scoring process can be then proceeded on the filtered phrase table.

Table 1. Algorithm for phrase filtering.

```
PhraseFilter($C, P$)
$P' \leftarrow P$
$V \leftarrow$ vocabulary of $C$
for each
do $p$ in $P'$
    if $p$ does not contain any word of $V$
        then Remove $p$ from $P'$
return $P'$
```

5 Experiment and Result

5.1 Data and Packages

The parallel data we used for training is listed in Table 2. We use NIST Open MT08 Current Test Set[2] as development data. And we use ICTCLAS[3] for Chinese word segmentation, Giza++ [6] for learning word alignment, SRILM [7] for the training of language model and MOSES [1] for decoding.

LDC number	Sentence pairs
LDC2004E12	4978744
LDC2002E18	109792
LDC2003E14	138884
LDC2005T10	156811
LDC2006E26	90699
total	5474930

Table 2. Statistic of training data.

5.2 Submitted Systems

All our submitted systems are made up of a single phrase translation model, reordering model and language model. The phrase translation model uses 5 features, including bidirectional conditional translation probabilities, lexical weights on both sides and a phrase penalty. A bi-directional reordering model was employed. The reordering probabilities are conditioned on lexicons of both sides. Reordering types include monotone, swap and discontinuous. The language model is a 3-gram model trained on GigaWord. The weights of these models are tuned on the development set using Minimum Error Rate Training[8].

We submitted one primary system and two contrast systems for the evaluation[4].

Primary: Original phrase-based system with no modification

Contrast1: Phrase-based system using filtered phrase table (as described in Section 4)

Contrast2: Phrase-based system using filtered and smoothed phrase table (as described in Section 4 and 3)

[2] LDC2009E09

[3] http://www.ictclas.org

[4] We did not submit our result using sentence segmentation due to some mistakes in experiments.

5.3 Post Processing

We perform a post translation using Chinese Name Entity translation dictionary[5] to translate the out of vocabulary (OOV) words. All those OOV words that not in the dictionary are translated to their Chinese Pinyin form.

5.4 Results and Analysis

System	Dev Score	Test Score
Primary	0.245775	0.1943
Contrast1	0.242960	0.1919
Contrast2	0.244143	0.1941

Table 3. System scores.

	Phrase table size (entries)
Primary	55.11M
Contrast1	3.63M

Table 4. Statistic of phrase table.

We can see from the results in Table 3 and 4 that filtering the phrase table leads to a little decrease of the final translation result. However, after filtering, the phrase table is reduced significantly, which will be much more convenient for applying other techniques. In our experiments, by phrase table smoothing, we can get comparable result with the original system.

6 Conclusion

This paper summaries our work for the NIST Open MT09 evaluation. We built our systems based on MOSES, and tried several methods to improve the result. Due to the limitation of time, we did not get good enough result from these methods. However, we find that some basic filtering techniques can largely reduce the size of phrase table with just a little decrease in BLEU score. The filtered phrase table may be more convenient and efficient for further use.

[5] LDC2005T34

Acknowledgement

This work is supported by the National High Technology Development 863 Program of China under Grants no. 2006AA010109, the National Natural Science Foundation of China under Grants no. 60673043 and the Scientific Research Foundation of Graduate School of Nanjing University under Grants no. 2008CL08.

References

1. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: HLT-NAACL. (2003)
2. Xu, J., Zens, R., Ney, H.: Sentence segmentation using ibm word alignment model 1. In: the 10th Annual Conference of the European Association for Machine Translation, Budapest, Hungary (May 2005) 280–287
3. Meng, B., Huang, S., Dai, X., Chen, J.: Segmenting long sentence pairs for statistical machine translation. In: International Conference on Asian Language Processing, Singapore (Dec 7-9 2009)
4. Chen, S.F., Goodman, J.T.: An empirical study of smoothing techniques for language modeling. Technical report, Computer Science Group, Harvard University, Technical Report TR-10-98 (1998)
5. Foster, G., Kuhn, R., Johnson, H.: Phrasetable smoothing for statistical machine translation. In: EMNLP. (2006)
6. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1) (2003) 19–51
7. Stolcke, A.: Srilm - an extensible language modeling toolkit. In: Proceedings of International Conference on Spoken Language Processing. (2002) 901 904
8. Och, F.J.: Minimum error rate training in statistical machine translation. In: ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2003) 160–167