The XMU Phrase-Based Statistical Machine Translation System for IWSLT 2006

Yidong Chen, Xiaodong Shi, Changle Zhou

Institute of Artificial Intelligence, School of Information Sciences and Technologies,
Xiamen University, Xiamen, Fujian, P. R. China
{ydchen, mandel, dozero}@xmu.edu.cn

Abstract

In this paper, an overview of the XMU phrase-based statistical machine translation system for the 2006 IWSLT Speech Translation Evaluation was given. In this year's evaluation, we participated in the open data track for ASR lattice and Cleaned Transcripts for the Chinese-English translation direction. The system ranked 7th among the 12 participating systems in the Chinese-English spontaneous speech ASR output task, 11th among the 15 participating systems in the Chinese-English read speech ASR output task and 8th among the 15 participating systems in the Cleaned Transcripts task.

1. Introduction

This paper describes the system which participated in the 2006 IWSLT Speech Translation Evaluation of Institute of Artificial Intelligence, Xiamen University. The system is a phrase-based Statistical Machine Translation (SMT) system. This paper is organized as follows. Section 2 describes data preparing. Section 3 gives an overview of the translation model. In section 4, experiments and the results are reported. And finally, section 5 concludes.

2. Preparing the Data

This section describes how we prepare the training data for our SMT system. Four steps are described in detail, that is., data preprocessing, word alignment, phrase extraction and phrase probabilities estimation.

2.1. Preprocessing

Data preprocessing is not a trivial task for machine translation system. Our experiments showed that good data preprocessing model can result in better translation quality.

Two types of preprocessing were performed on the Chinese part of the training data:

- Segmentation: To transform Chinese characters into Chinese words.
- SBC case to DBC case: To replace numbers, English characters or punctuations in SBC case in Chinese by their DBC case. For instance, "１", "Ａ" and "。" would respectively be replaced by "1", "A" and ".".

For the English part of the training data, also two types of preprocessing were performed:

- Tokenization: To separate punctuations from words in English sentences.
- Truecasing of the first word of an English sentence: To transform the uppercase version of the beginning words of English sentences into their lowercase version if their lowercase version occur more often.

2.2. Word Alignment

To achieve n-to-n word alignment, we first run GIZA++ up to IBM model 4 in both translation directions to get an initial word alignment, and then apply "grow-diag-final" method [1] to refine it. This process could be addressed in detail as followed:

- In the *initial* step, we intersect the two alignments obtained by running GIZA++, i.e., Chinese to English and English to Chinese, and get a high-precision alignment.
- Then the intersection alignment *grows* iteratively by adding potential alignments, which exist in the union of the two alignments. The neighbors of the intersection points in alignment matrix, including left, right, up, bottom and the diagonally directions are checked, if either of the words linked by the potential alignment is not aligned previously, the potential alignment is added. This operator is done until no more neighbors can be added.
- In the *final* step, potential alignments, which exist in the union of two alignments, will be added if all their neighbors do not exist in the union alignment.

2.3. Phrase Extraction

Bilingual phrases can be learned from word aligned parallel corpus. As is common in most phrase-based SMT systems, we consider bilingual phrase as a pair of source and target words sequences, with the following constrains:

- The words should be consecutive in both source and target sentences.
- The word level alignment of bilingual phrase should be consistent with the alignment matrix.

The consistency means that the words of the bilingual phrase can only be aligned to each other, and not to any other words outside.

Our phrase extraction method is very similar to [2]. For a word aligned sentence pair, we enumerate all the consecutive words sequences of English sentence, and for each English phrase, find the corresponding Chinese words according to the alignment matrix, if it satisfies the two constraints above, a bilingual phrase is extracted. In addition, in order to extract

more phrases, such a bilingual phrase can be extended at Chinese side since "NULL" alignment is allowed, which means a word aligned to nothing. For the same English phrase, we extend the corresponding Chinese phrase to both left and right, if the added Chinese word is not aligned, and the new phrase satisfies our definition, it is extracted as a bilingual phrase. This is done iteratively until no more words can be added.

However, we limited the length of phrases from 1 word to 6 words in our experiment, since it has been showed that longer phrases don't yield better translation quality [1]. And, to avoid a too large search space in decoding, we also limited the size of the translation table. For a Chinese phrase, only 20-best corresponding bilingual phrases were kept. We used Formula 1 to evaluate and rank the bilingual phrases with the same Chinese phrase.

$$\sum_{i=1}^{N} \lambda_i \cdot h_i(\tilde{e}, \tilde{c}) \tag{1}$$

Where, $h_i(\tilde{e}, \tilde{c})$ ($1 \le i \le N$) denotes a phrase probability of a given bilingual phrase $(\tilde{e}, \tilde{c})$, and λ_i ($1 \le i \le N$) is the corresponding parameter for $h_i(\tilde{e}, \tilde{c})$. In our system, N is set to be 4, in that there are four phrase probabilities for a given bilingual phrase (see 2.4 for details).

Note that, the parameters here should use the same values as their corresponding ones in the translation model (see 3.2 for details).

By using the pruned phrase table, our system could translate the test set from this evaluation at the speed of about 0.2 seconds per sentence.

2.4. Phrase Probabilities

Four phrase probabilities are defined for a given bilingual phrase in our system:

- Phrase translation probability $p(\tilde{e} \mid \tilde{c})$
- Inversed phrase translation probability $p(\tilde{c} \mid \tilde{e})$
- Phrase lexical weigh $lex(\tilde{e} \mid \tilde{c})$
- Inversed phrase lexical weight $lex(\tilde{c} \mid \tilde{e})$

We define the phrase translation probability using relative frequency as in Formula 2:

$$p(\tilde{e} \mid \tilde{c}) = \frac{N(\tilde{e}, \tilde{c})}{\sum_{\tilde{e}'} N(\tilde{e}', \tilde{c})} \tag{2}$$

Where, $N(\tilde{e}, \tilde{c})$ is the total number of bilingual phrase $(\tilde{e}, \tilde{c})$ occurred in the training corpus.

Additional to $p(\tilde{e} \mid \tilde{c})$, we introduce a lexical weight metric that denotes how well the words of phrase $\tilde{c}$ translate to the words of phrase $\tilde{e}$. Following the description in [1], given a bilingual phrase (e_1^I, c_1^J) and its alignment a, the lexical weight is defined as Formula 3:

$$lex(e_1^I \mid c_1^J, a) = \prod_{i=1}^{I} \frac{1}{|\{j \mid (i,j) \in a\}|} \sum_{\forall (i,j) \in a} p(c_i \mid e_j) \tag{3}$$

For computing phrase lexical weight, we should know the word level alignment of bilingual phrases, as well as the word translation probability. When extracting phrases from the training corpus, the alignment information is reserved, moreover, a special token "NULL" is added to each English sentence and aligned to unaligned Chinese words, and then the word translation probability can be computed using relative frequency.

The inversed phrase translation probability $p(\tilde{c} \mid \tilde{e})$ and the inversed phrase lexical weight $lex(c_1^I \mid e_1^J, a)$ can be computed in the similar way to $p(\tilde{e} \mid \tilde{c})$ and $lex(e_1^I \mid c_1^J, a)$, respectively.

3. System Overview

This section gives an overview of our system, including the translation model and the search algorithm. We also introduce the preprocessing model we developed to recover the missing punctuations of Chinese sentences in the test set and the way we used to translate the ASR lattice.

3.1. Translation Model

As described in [3], we use a log-linear modeling approach, in which all knowledge sources are described as feature functions that include the given source language string c_1^J and the target language string e_1^I. Hence, the translation probability and the decision rule could be given by Formula 4 and 5, respectively.

$$\Pr(e_1^I \mid c_1^J) = \frac{\exp[\sum_{m=1}^{M} \lambda_m \cdot h_m(e_1^I, c_1^J)]}{\sum_{e_1^I} \exp[\sum_{m=1}^{M} \lambda_m \cdot h_m(e_1^I, c_1^J)]} \tag{4}$$

$$\hat{e}_1^I = \arg\max_{e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m \cdot h_m(e_1^I, c_1^J) \right\} \tag{5}$$

Seven features were used in our translation model:

- Phrase translation probability $p(\tilde{e} \mid \tilde{c})$
- Inversed phrase translation probability $p(\tilde{c} \mid \tilde{e})$
- Phrase lexical weigh $lex(\tilde{e} \mid \tilde{c})$
- Inversed phrase lexical weight $lex(\tilde{c} \mid \tilde{e})$
- English language model $lm(e_1^I)$
- English sentence length penalty I
- Chinese phrase count penalty $-J'$

Note that, Features on reordering were not yet taken into consideration in our model. We hope to consider the reordering problem carefully and integrate such features in the model in the future.

3.2. Parameters

The parameters used in the translation model could be trained using discriminative training method such as minimum error rate training [4].

But due to the time limitation, we didn't implement such method. So we have to adjust the parameters by hand. Moreover, we didn't readjust the parameters according to the develop sets provided in this evaluation again due to the time limitation. On the contrary, we simply used an empirical setting, with which our decoder achieved a good performance in translating the test set from the *2005 China's National 863 MT Evaluation*. The parameter settings for our system are listed in Table 1, as followed:

Table 1: The parameter settings

Parameters	Corresponding Features	Values
λ_1	$p(\tilde{e} \mid \tilde{c})$	0.15
λ_2	$p(\tilde{c} \mid \tilde{e})$	0.03
λ_3	$lex(\tilde{e} \mid \tilde{c})$	0.16
λ_4	$lex(\tilde{c} \mid \tilde{e})$	0.03
λ_5	$lm(e_1^I)$	0.13
λ_6	I	0.48
λ_7	$-J$	0.48

Please note that the parameter settings listed above is not optimal for the training and test set from this evaluation.

3.3. Decoder

We used the monotone search in the decoding, as described in [5]. And the monotone search was implemented with dynamic programming.

For the maximization problem in Formula 5, we define the quantity $Q(j,e)$ as the maximum probability of a phrase sequence. Thus $Q(J+1,\$)$ is the probability of the optimal translation, where the \$ symbol is the sentence boundary marker. Given the definitions, we then obtain the following dynamic programming recursion:

$$Q(0,\$) = 1 \tag{6}$$

$$Q(j,e) = \max_{\substack{0 \le j' < j \\ e', \tilde{e}}} \left\{ Q(j',e') + \sum_{m=1}^{M} \lambda_m \cdot h_m(\tilde{e}, c_{j'+1}^{j}) \right\} \tag{7}$$

$$Q(J+1,\$) = \max_{e'} \left\{ Q(J,e') + p(\$ \mid e') \right\} \tag{8}$$

During the search, we stored back-pointers to the maximizing arguments. So after performing the search, we could generate the optimal translation, easily.

3.4. Dealing with the Unknown Words

Words that are not covered by phrases are called unknown words. Keeping unknown words un-translated will make the translations less readable, so most phrase-based systems integrated a model to deal with them. Some systems simply dropped the unknown words [6] while other systems integrated a pre-translation model to detect and translate special unknown words such as named entities and simply dropped other unknown words [7].

In our system, no special translation models for named entities are used. Named entities are translated in the same way as other unknown words. During the decoding, an unknown word will be translated in two steps, as followed:

- Firstly, we will look up a dictionary containing more than 100,000 Chinese words for the word. All the translations will be put into the phrase table with a certain probability, and the most optimal one will be selected by the translation model. In this evaluation, the probability was set to be 10^{-7}.
- If no translations are found in the first step, the word will then be translated using a rule-based Chinese-English translation system[1].

Using the steps described above, all the 63 unknown words in the test data for the Cleaned Transcripts task in this evaluation are translated into English.

3.5. Recovering the Missing Punctuations

One of the differences between the test data of this year's evaluation and those of the previous years' is that there are no punctuations in the Chinese sentences this year. The missing of punctuations can have an adverse effect on the translation quality, so we developed a preprocessing model to recover the missing punctuations.

Given a Chinese sentence with N words, $w_1\ w_2\ \ldots\ w_N$, we may construct a directed graph with $2N$+1 levels, as showed in Figure 1.

In Figure 1, $level_0$, $level_{2N}$ and $level_{2i-1}$ ($1 \le i \le N$) all contains one node, while $level_{2i}$ ($1 \le i < N$) contains M nodes. The node in $level_0$ and $level_{2N}$ are both labeled \$ and is used to represent the start sentence boundary and the end sentence boundary respectively. The node in $level_{2i-1}$ is used for ith word w_i ($1 \le i \le N$). And nodes in $level_{2i}$ ($1 \le i < N$) are used for all possible punctuations. Here at most M kinds of punctuations are taken into consideration.

Given such a graph, the problem of punctuation recovering could be looked on as a problem of searching the optimal path from the node in $level_0$ to the node in $level_{2N}$. In this problem, a path is said to be better than the other one if the language model score for it is larger than that for the latter. We then used the Viterbi algorithm [8] to solve the search problem.

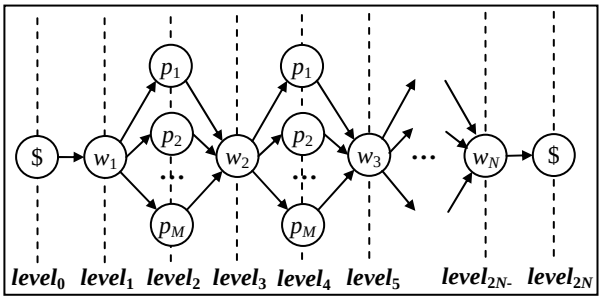

Figure 1: Search graphic for punctuation recovering.

[1] Downloadable from http://59.77.17.146/download/

3.6. Translating the ASR Lattice

In the task of translating the ASR lattice, three types of test data were given:

- word lattice
- the 20-best results generated from ASR lattice
- the 1-best results generated form ASR lattice

Due to the time limitation, we finally completed this task using a simpler approximate way:

- We first used our system to translate all the 20-best results and got 20 translations for each corresponding sentence.
- Then we used the English language model to choose the best translation for each sentence.

According to our observation, for many test sentences, all the results in the 20-best set contain mistakes. So the method described above is not a very good solution. A possibly more reasonable way is to regenerate the 1-best result based on Chinese language model from word lattice and then to translate it. We will try this idea in the future.

4. Experiments

In this year's evaluation, we participated in the open data track for ASR lattice and Cleaned Transcripts for the Chinese-English translation direction.

This section describes the training data we used and the results we achieved. Some discussions follow the results.

4.1. Training Data

We participated in the open data track this year. Because in addition to the training data provided by IWSLT 2006, we also used other training data.

All the data we used were list in Table 2.

Table 2: Training data list

Purposes	Corpus	
	Names	Amounts
Bilingual Phrase	Training set from IWSLT 2006	39,952 sentence pairs
	Training set from the 2005 China's National 863 MT Evaluation	152,049 sentence pairs
English Language Model	English part of the training set from the 2005 China's National 863 MT Evaluation	7.4M words
Chinese Language Model	Chinese part of the raining set from IWSLT 2006	350K Chinese words
	Chinese Reader (Duzhe) Corpus	7.9M Chinese words

We use SRI Language Modeling Toolkit [9] to train language model with modified Kneser-Ney smoothing [10]. Only trigram language model was trained on the training corpus.

The use of additional data did help improving the performance of our system on the develop sets. Especially, the use of additional bitexts gained about 0.06 absolute improvements in blue-4 score for develop set 1, 0.07 for develop set 2, 0.05 for develop set 3 and 0.03 for develop set 4. So we included more training data in the evaluation.

4.2. Results

The scores of our system in IWSLT 2006 is list in Table 3, only the BLEU-4 scores are included.

Table 3: BLEU-4 scores for Xiamen-U in IWSLT 2006

	official (with case + punctuation)	additional (without case + punctuation)
CE spontaneous speech ASR output	0.1505	0.1623
CE read speech ASR output	0.1579	0.1718
Correct Recognition Result	0.1976	0.2162

Some lessons could be learned from the scores in Table 3:

- The scores on Correct Recognition Result are significantly higher than those on ASR output. This may result from the influence of the ASR errors. And the other reason may be the simple method we used to translate ASR lattice.
- The scores on CE read speech ASR output are slightly higher than those on CE spontaneous speech ASR output. This indicates that the ASR system used to give the ASR output may be cleverer at the read speech data than at the spontaneous speech data.
- The additional scores are higher than the official scores. This indicates that post-editing models such as truecasing or punctuation correction may help improving the translation quality. We will integrate such models in the future.

5. Conclusions

This paper describes the system which participated in the 2006 IWSLT Speech Translation Evaluation of Institute of Artificial Intelligence, Xiamen University. It is a rather crude phrase-based SMT baseline, for example, without even considering phrase reordering. More improvements are underway.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 60573189 and Grant No, 60373080), National 863 High-tech Program (Grant No

2004AA117010-06) and the Fund of Key Research Project of Fujian Province (Grant No 2006H0038)

7. References

[1] Koehn, Philipp, Och, Fraz Josef and Marcu Danie, "Statistical phrase-based translation", *Proceeding of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Edmonton, Canada, 2003, pp. 127-133.

[2] Och, Franz Josef, "Statistical Machine Translation: From Single Word Models to Alignment Templates", *Ph.D. thesis*, RWTH Adchen, Germany, 2002.

[3] Och, Fraz Josef and Ney, Hermann, "Discriminative training and maximum entropy models for statistical machine translation", *Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, 2002, pp. 295-302.

[4] Och, Franz Josef, "Minimum error rate training in statistical machine translation", *Proceeding of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003, pp. 160-167.

[5] Zens, Richard, Och, Franz Josef and Ney, Hermann, "Phrase-Based Statistical Machine Translation", *Proceeding of the 25th German Conference on Artificial Intelligence (KI2002)*, ser. *Lecture Notes in Artificial Intelligence (LNAI)*, M. Jarke, J. Koehler, and G. Lakemeyer, Eds., Vol. 2479. Aachen, Germany: Springer Verlag, September 2002, pp. 18–32.

[6] Koehn, Philipp, Axelrod, Amittai, Mayne, Alexandra Birch, Callison-Burch, Chris, Osborne, Miles and Talbot, David, "Edinburgh system description for the 2005 iwslt speech translation evaluation", *Proceeding of International Workshop on Spoken Language Translation*, Pittsburgh, PA, 2005

[7] He, Zhongjun, Liu, Yang, Xiong, Deyi, Hou, Hongxu and Liu, Qun, "ICT System Description for the 2006 TC-STAR Run #2 SLT Evaluation", *Proceeding of the TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, 2006, pp. 63-68.

[8] Forney, G. D., "The Viterbi algorithm", *Proceeding of IEEE*, 61(2): 268-278, 1973

[9] Stolcke, Andreas, "Srilm – an extensible language modeling toolkit", *Proceedings of the International Conference on Spoken language Processing*, 2002, volume 2, pp. 901–904.

[10] Chen, Stanley F. and Goodman, Joshua, "An empirical study of smoothing techniques for language modeling", *Technical Report TR-10-98*, Harvard University Center for Research in Computing Technology, 1998.