Some Improvements in Phrase-Based Statistical Machine Translation

Zhendong Yang [1], Wei Pang [1], Jinhua Du[1], Wei Wei[1], Bo Xu[1]

[1] Hi-tech Innovation Center, Institute of Automation
Chinese Academy of Sciences, 100080 Beijing
{ zdyang, wpang, jhdu,weiwei, xubo}@hitic.ia.ac.cn

Abstract. In statistical machine translation, many of the top-performing systems are phrase-based systems. This paper describes a phrase-based translation system and some improvements. We use more information to compute translation probability. The scaling factors of the log-linear models are estimated by the minimum error rate training that uses an evaluation criteria to balance BLEU and NIST scores. We extract phrase-template from initial phrases to deal with data sparseness and distortion problem through decoding. By re-ranking the n-best list of translations generated firstly, the system gets the final output. Some experiments concerned show that all these refinements are beneficial to get better results.

Keywords: phrase-based translation, minimum error rate training, phrase-template, re-scoring.

1 Introduction

Statistical machine translation is a promising approach to large vocabulary text translation. Inspired by the Candide system IBM developed in the early 1990s [1], many statistical machine translation systems have been proposed. From the word-based system initially, phrased-based and syntax-based translation systems have been developed [2][3].

We have proposed a phrase-based translation system [4]: In the system, we applies phrase-based translation model to capture the corresponding relationship between source and target language. A phrase-based decoder we developed employs a beam search algorithm, in which some target language words that have both high frequency of appearance and also fertility zero are introduced to make the result more reasonable. We improve the previously proposed tracing back algorithm to get the best path.

This paper shows some improvements of our system currently: Section 2 presents the architecture of our system. Section 3 describes how to extract phrase-template from initial phrases. Section 4 studies the approach to compute the translation probability and train the scaling factors of all the models used in the translation system. Our system uses some special information to re-score the n-best translations,

this is outlined in Section 5. In Section 6, a series of experiments are presented. We analyze the results. We summarize our system in Section 7.

2 System Description

In statistical machine translation, we are given a source language (Chinese) sentence $c_1^J = c_1 \cdots c_j \cdots c_J$, the goal is to generate the target language (English) sentence $e_1^I = e_1 \cdots e_i \cdots e_I$ which maximize the posterior probability:

$$\begin{aligned} e_1^I &= \arg\max_{e_1^I} \{\Pr(e_1^I \mid c_1^J)\} \\ &= \arg\max_{e_1^I} \{\Pr(e_1^I)\Pr(c_1^J \mid e_1^I)\} \end{aligned} \quad (1)$$

Applying the maximum entropy framework [5], the conditional distribution $\Pr(e_1^I \mid c_1^J)$ can be modeled through suitable feature functions, our system is based on a log-linear model which extends the word-based IBM Model to phrase-based model. We obtain:

$$\begin{aligned} e_1^I &= \arg\max_{e_1^I} \{\Pr(e_1^I \mid c_1^J)\} \\ &= \arg\max_{e_1^I} \{\exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, c_1^J))\} \\ &= \arg\max_{e_1^I} \{\sum_{m=1}^{M} \lambda_m h_m(e_1^I, c_1^J)\} \end{aligned} \quad . \quad (2)$$

Our system uses some feature models to drive the translation process: translation model, language model, distortion model and future score model. The system exploits two search passes: the first is performed by a beam search [4] to obtain n-best results, the second is a re-scoring algorithm to get the final output. The process is illustrated as follows:

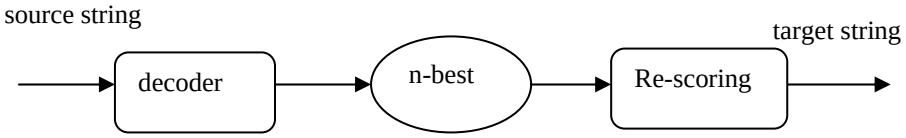

Fig. 1. The decoding illustration of our phrase-based translation system

3 Generalizing phrases

The translation system often encounters data sparseness and distortion problem. Koehn et al. [6] find data sparseness takes over for long phrases. Many systems use very simple distortion model [6][7] to reorder phrases, which penalizes translations according to the jump distance instead of the syntax information. This model takes the risk of dealing with the long distance jump that is usually common between two languages with different expression habit. So we extract phrase-template from initial phrase pairs to alleviate these problems. The phrase are generalized in two ways:

If the phrase pairs include named entities such as named persons, locations and organization, or numeral, we replace them with some certain symbols in source and target sides. We call the generalized phrase of this type N_template. Some symbol examples are showed in table 1. Name entities are translated by separate rule-based module through searching process.

Table 1. Some symbol examples replacing name entities

type	Symbol in source side	Symbol in target side
named persons	PER_	per_
locations	LOC_	loc_
organizations	ORG_	org_
numeral	TIMP_	timp_

Besides this method, we generalize phrase pairs that don't include named entities as done in [8], we call the rule generated X_template :

1. Initial phrase pairs are extracted from sentences similar to [6]. Every initial phrase pair is an X_template.
2. Aligned source-target small phrase included in initial phrases can be replaced by a nonterminal X. Then the phrase-template is extracted.
3. We only extract phrase template that has only one nonterminal and at least one terminal. This prevent from producing too many phrase-template.

The generalized phrases have some forms like that in table 2.

Because many phrase pairs may generate the same X_template. We select the highest translation probability of these phrase as that of X_template. But its real translation probability is the product of probability of X_template and the relevant nonterminal X. During decoding, only when the phrase can not be found in the phrase table, the corresponding phrase-template is used.

Table 2. The forms of phrase-template generated

Type	Initial phrase pairs (source # target)	Generalized phrase pairs (source # target)
N_tempate	9 点到达 # get at 9	TIMP_到达 # get at timp_
	从 济南 到 武汉# from Jinan to Wuhan	从 LOC_到 LOC_ # from loc_ to loc_
X_template	与选民 建立 关系 # connect with the voters	与 X_1 建立 关系 # connect with X_1
	他的 朋友 之一 # one of his friends	X_1 之一 # one of X_1

4 Translation model and Minimum Error Rate Training

There are several approaches to compute the phrase translation probability, Koehn estimate the probability distribution by relative frequency [6]:

$$h_1 = \frac{c(\tilde{c},\tilde{e})}{c(\tilde{c})} \quad . \tag{3}$$

or

$$h_2 = \frac{c(\tilde{c},\tilde{e})}{c(\tilde{e})} \quad . \tag{4}$$

Where $\tilde{e}$ and $\tilde{c}$ are English and corresponding Chinese phrase, and c(•) means the occurrence count in the training data. But if two phrase pairs have the same frequency, the probabilities have little discrimination. To get more discriminative probability, CMU calculate probabilities based on a statistical lexicon (such as IBM model 4) for the constituent words in the phrase, the formula is :

$$h_3 = \prod_i \sum_j p(c_i \mid e_j) \tag{5}$$

or inverse the formula:

$$h_4 = \prod_j \sum_i p(e_j \mid c_i) \quad . \tag{6}$$

where c_i and e_j are the words that constitute phrase $\tilde{c}$ and $\tilde{e}$, $p(c_i \mid e_j)$ is the IBM model. This method has a drawback: If only one word of source phrase has no appropriate corresponding word in target phrase, the phrase translation probability will be small.

In order to offset the shortcoming of each method, we combine these four formulas to compute the phrase translation probability. The four formulas, distortion model, language model and future model are combined by log-linear form with a scaling factor each. The factors are estimated on the development data, by applying a minimum error training procedure [9]. It is an optimization problem, we use the simplex algorithm [10] to solve this problem. A key role of this training process is the evaluate metric. Firstly, we select BLEU as metric, we get a high BLEU score, but a low NIST score because the output sentences are short. Accordingly, we get a high NIST score at the cost of a significant deterioration of BLEU score when the NIST is used as the evaluate metric. A reasonable trade-off was final acquired using the metric:

100*BLEU + 5*NIST

The coefficient training process are introduced as follows:

1 Give every model scaling factor an initial value.

2 Use the current factor value to obtain the n-best candidate translations and corresponding features for each sentence through decoding. Merge the n-best lists across iterations.

3 Run the minimum error training to get the factor value of this iteration. If the value converges, the process stops, otherwise, goes to 2. The maximum of iteration is set as 10.

5 Re-scoring

The output sentence with the highest probability sometimes is not the best one compared with the reference translation. So we apply three additional feature functions to re-rank each of the 500 candidate translations for every input sentence:

- 2-gram target language model..
- 4-grams target language model

• Question feature, that is, if the input sentence ends with a question punctuation, we alleviate the penalizing on distortion.

• Name entity feature, i.e. if the number of name entity of output sentence is equal to that of source sentence, a binary feature is triggered to favor this translation.

We use the SRI Language modeling Toolkit to train language model. These features can be used respectively or combinatorially. We first get the top 500 candidates for every input sentence, then re-rank them to obtain the final output. The experiments are introduced in Section 6.

6 Experiments

We carry a number of experiments on 2005 863 Chinese-to-English MT evaluation tasks of China. 870,000 sentence pairs are used as training data to train the translation and language model. 500 Chinese sentences with about 4 reference translation sentences each are used as development data, we use the development data to optimize the model scaling factors. About 450 sentences are reserved for testing all the experiments. All these data are from the 2005 863 MT evaluation data. These sentences are about tour and daily life with the length of 5-20 words.

First we do experiments on the test data to check the role of the phrase-template. The experiments are made without training the model scaling factor. The results are shown in figure2. Where No_template denotes no phrase-template used, and +_template denotes adding phrase-template. We can see with the phrase-template added, the BLEU score goes up from 0.182 to 0.197, NIST score increases from 4.77 to 5.86. This experiment shows the phrase-templates play a positive role because they partly remedy the data sparseness and distortion problem. So we train the model factor by minimum-error-rate training with phrase-template added. The results are showed in figure 3 and table 3.

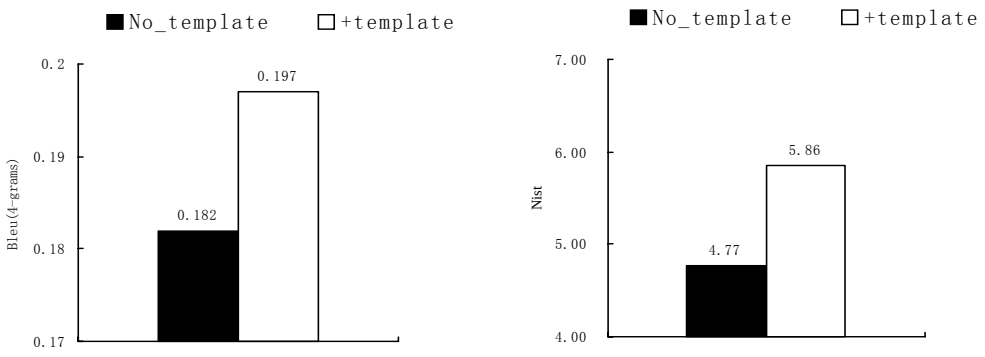

Fig. 2. The role of the phrase-template

We make minimum error rate training on development set. We translate the sentence of test data to check the effect of the training, the results are showed in table 3. From figure 3, we can see the BLEU score' changing trend with the total number of translation candidates. The NIST score changes like this situation. The training procedure is iterated until the n-best list remains stable. In our experiment, about 9 iterations are needed for convergence. The final values of each model' scaling factor are showed in table 3. The BLEU score increases 0.015 from 0.197 to 0.212, and the NIST score goes up from 5.86 to 6.22.

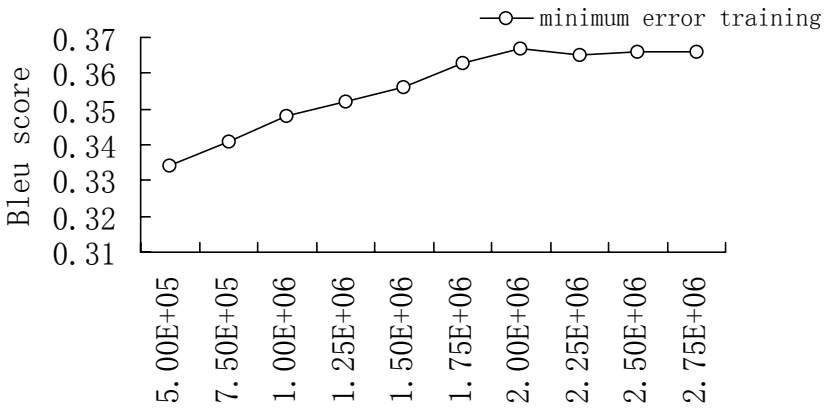

Fig.3. Blue score as a function of the total number of generated translation candidates.

Table .3. The final model factors and the BLEU and NIST score results when translating the test data. $\lambda_1 \sim \lambda_4$ mean the translation model factor, λ_{Lm}, λ_{dis} and λ_{fut} mean the scaling factor of language model, distortion modle and future score model respectively.

λ_1	λ_2	λ_3	λ_4	λ_{Lm}	λ_{dis}	λ_{fut}
1.0628	0.0031	2.4476	0.7702	1.3992	0.1185	1.6576
BLEU (4-gram)		0.212				
NIST		6.22				

Finally, we do experiments on the test set for re-ranking. Table 4 shows the single contribution of the 4 feature functions used. Almost all the features enhance the performance except question feature, this is because the phrase-template has partly resolve reordering phrases. This feature breaks the balance of all the models used. The result from 4-gram feature is superior to other methods, this indicates the n-gram feature provides a significant role on fidelity and fluency of the translation. Combining these methods always leads to some improvement.

Table.4. The effect of each feature functions in re-scoring step on the test data.

System	BLEU(4-gram)	NIST
baseline	0.212	6.22
Question feature	0.202	5.92
2-grams LM	0.213	6.31
4-grams LM	0.221	6.64
Name entity feature	0.216	6.52
All features	0.224	6.83

7 Conclusion

In summary, this paper shows some improvements to our phrase-based translation system. We use phrase-template to alleviate data sparseness and reorder the phrases during translation. The translation model is refined, and the scaling factors of the all the models are estimated by minimum error rate training. Instead of output the translation with the highest probability, we re-score the n-best lists to get the final translation. All these efforts are effective to our system.

Although we used some formal syntax to generalize the phrases, how to combine syntax with phrase is our important work next. We will do some studies about parsing to improve our phrase-based system next step.

References

1. Peter F. Brown , Stephen A. Della Pietra, Vincent J. Della Pietra, and Pobert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, vol. 19, no. 2, (1993), pp. 263-311,.
2. Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venugopal, Bing Zhao, Alex Waibel . The CMU Statistical Machine Translation System. In proceedings of the Ninth Machine Translation Summit.(2003).
3. Yamada, K. and Knight. A Syntax-based Statistical Translation Model. In Proc. of the 39th Annual Meeting of ACL, (2001).
4. Zhendong Yang, ZhenBiao Chen, Wei Pang, Wei Wei, Bo Xu, The CASIA Phrase-Based Machine Translation System, IEEE NLPKE'05, Wuhan, China, (2005),pp.416-419.
5. A Berger, S. Della Pietra, and .Della Pietra, "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics, vol. 22, no.1, (1996), pp 39-71.
6. Koehn, P. ,Och, F. J., and Marcu , D. Statistical Phrase-Based Translation. In Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics. (2003).
7. Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. Computational Linguistics, 30:417-449.
8. David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In Proceedings of ACL 2005.pages 263-270.

9. Franz Josef OCH. Minimum Error Rate Training in statistical machine translation . In Proc. of the 41th Annual Meeting of the Association of Computational Linguistics(ACL), Sapporo, Japan.2003
10. William H. PRESS, Saul TEUKOLSKY, William T. VETTERLING and Brian P. FLANNERY. 2002. Numerical Recipes in C++. Cambrige University Press, Cambridge, UK.