The CASIA Phrase-Based Machine Translation System

Wei Pang, Zhendong Yang, Zhenbiao Chen, Wei Wei, Bo Xu, Chengqing Zong
Institute of Automation Chinese Academy of Sciences
wpang@hitic.ia.ac.cn

Abstract

In this paper we propose a phrase-based translation system. In the system, we use phrase translation model instead of word-based model. An improved method of computing phrase translation probability is studied. We translate numeral phrases first by using a standard templates depository. We develop a phrase-based decoder that employs a beam search algorithm. To make the result more reasonable, we apply those words with fertility probability of zero. We improve the previously proposed tracing back algorithm to get the best path. Some experiments concerned are presented.

1 Introduction

Statistical machine translation is a promising approach for large vocabulary text translation. In the early 90s, IBM developed Candide system. Since then, many statistical machine translation systems were proposed [2][3]. These systems apply a translation model to capture the relationship between the source and target languages, and use a language model to drive the search process. The primary IBM model was purely word-based. To get more complex structure, better lexical choice and more reliable local reordering, the phrase-based statistical machine translation systems were proposed. Yamada and Knight [4] used phrase translation in a syntax-based translation system; March and Wong [5] introduced a joint-probability model for phrase translation; CMU and IBM also improved their systems with phrase translation capability.

Our system applies a phrase-based translation model to capture the corresponding relationship between two languages. We propose a formula to compute the phrase translation probability through word alignment. The phrase-based decoder we developed employs a beam search algorithm, similar to the one in [6]. We applied zero fertility words in the target language. Because the translation quality largely depends on the accuracy of phrase-to-phrase translation pairs extracted from bilingual corpora, we propose a different tracing back algorithm to find the best path. Four methods are studied to extract bilingual phrase pairs. We describe these methods and phrase-based translation model in Section 2. Section 3 explains the method of numeral phrase translation. Section 4 outlines the architecture of the decoder that combines the translation model, distortion model, language model to generate target sentence. In Section 5, we present a series of experiments in which Chinese sentences are translated into English sentences, and analyze the results of these experiments. We make a summary in Section 6.

2 Phrase Translation Model

Our system is based on a phrase translation model, which is different from the original IBM model. The phrase we mention here is composed of a series of words that perhaps possess no syntax or semantic meanings. In addition, a word can also be treated as a phrase, so the word-based model is included in the phrase-based model.

There are different methods of getting phrase pairs from a bilingual corpus .We used four methods as follows:

2-1 Extracting directly through IBM Word-Based model

By using IBM model 4, we get a series of target language words that correspond to the source language words of the bilingual sentence pair. Then we form these words into the target phrase. For example, phrase translation $(\tilde{c}, \tilde{e})$ is selected from a bilingual sentence pair (c,e). $\tilde{c} = c_1 c_2 \cdots c_i$, where $c_1 c_2 \cdots\cdots$ are the words form the source language phrase. According to IBM model 4, each word c_i of phrase $\tilde{c}$ can find its correspondent target language word in e with a certain probability, and get the target word's position in e. If there are more than one target language word correspond to c_i, then the one with the highest probability is selected. Extracting the words that lie between the minimum and maximum position, we get the correspondent target phrase $\tilde{e} = e_1 e_2 \cdots e_j$. This method is rather simple, but the length of target phrase and source phrase may differ greatly. So we can set a threshold of length or translation probability to make the result more reasonable.

2-2 Integrated Segmentation and Phrase Alignment (ISA)

In training corpus, each sentence pair (F,E) is represented as a two-dimensional matrix R_{n*m}. f is composed of n words (f_1, f_2……f_n), and e is made up of m words (e_1, e_2……e_m). To measure the "goodness" of translating a source word to a target word, we use the value of Point-wise Mutual Information (MI) between these two words. Thus, we can mark the value (e,f) in the matrix as I(e,f) ,

$$I(e,f) = \log_2 \frac{P(e,f)}{P(e)P(f)} \qquad (2\text{-}1)$$

The value of P(e), P(f), P(e,f) can be numerated from the training result. With all nodes value computed in the matrix, we can get an MI matrix of a sentence pair.

We extract the phrases as the following steps:

a. Select a point with the highest value in matrix, and mark it as max(i,j).

b. Confine it with an evaluating function (for example, the ratio of two nodes' value in the matrix, $I(f_{x1},e_{y1})/ I(f_{x2},e_{y2})>num$).

c.Expand the 'max' cell to the largest possible rectangle regions (R_{start}, R_{end}, C_{start},C_{end}) under two constraints: (1). all the cells in the expanded region accord with the evaluation function ; (2). all the cells should not be marked.

d. The words in this area make up of a phrase pair. Mark all nodes between x-coordinate and y-coordinate in this matrix, then search other max points and corresponding rectangles among the rest unmarked nodes until all nodes in the MI matrix of this sentence pair are marked.[7]

2-3 Extracting Phrase Pairs From HMM Word Alignment Model

A simple way to extract phrase pairs is using a word alignment model. We use the HMM-based alignment model introduced in [8]. For a source phrase that ranges from position j_1 to j_2 in sentence, we can get the corresponding target phrase's beginning position and ending position to extract the phrase translation. Just like the method described in 2-1, a given factor that prevents the length of the phrase pairs differ greatly is needed.

2-4 Extracting phrase pair by Giza++ toolkit

The Giza++ toolkit can be used to establish word-based alignments. There are some heuristic functions can improve the quality of alignment and extract phrase pair. In [6], the parallel corpus is aligned bidirectionally, some additional alignment points are added to the intersection of the two alignments. All aligned phrase pairs are connected to be consistent with the word alignment: each word corresponds strictly to another word in a legal phrase pair, not to any word outside the pair [9].

2-5 Phrase Translation Probability

CMU used the phrase translation probability formula based on the IBM1 alignment model:

$$p(\tilde{c} \mid \tilde{e}) = \prod_i \sum_j p(c_i \mid e_j) \quad (2\text{-}2)$$

There is a drawback for this method: If only one word of source phrase has no appropriate corresponding word in target phrase, the phrase translation probability will be small. Since there are many auxiliary words and mood words in Chinese, this issue becomes more serious. To prevent this, we use the word alignment generated by the IBM model 4 to divide the whole phrase pair into several small phrase pair blocks. If one source word aligns to several target words or several source words align to one target word, then they are selected to form a block. Thus, the phrase translation probability formula becomes:

$$p(c \mid e) = \prod_i \left(\frac{1}{n_i} \sum_k \sum_j p(c_{ik} \mid e_j)\right) \quad (2\text{-}3)$$

where i is the sequence number of the small phrase translation blocks, k is the sequence number of the words in the i phrase block, j is the sequence number of the target word in the phrase, and n_i is the total number of words in block i.

3. Numeral

We can always find numeral in translation. In Chinese, the express method of numeral is rather simple. While in English, it’s more complicated, which makes several possible translation results from Chinese to English. In our system, numeral are picked out for special treatment to reduce mistakes in translation.

We summarized 5 ways of numeral translation: number translation, count translation, ordinal translation, year translation, and rule translation.

Number translation: For phone numbers and room numbers, they are only cardinal numbers. We can just translate them directly from Chinese to English.

Count translation: For integers, they are entirely constituted by numbers, and they may have digit numeral ,such as “百”. In Chinese, we count numbers by 4 digits. While in English, we count numbers by 3 digits. So it’s inappropriate to translate them directly. We adopted Arabic Numerals as an intermediary in translation.

Ordinal translation: For ordinal numbers, we can just translate them by corresponding English ordinal numbers.

Year translation: Divide it into 2 two-number, then translate accordingly.

Rule translation: Some numeral are made up of numbers and other words, and numbers only mean some sequence. In translation, there would be no numbers in English, such as Monday, March, and so on.

For these numeral phrase extracted from the training materials, we build up a template depository. Each pair of templates include a template of Chinese numeral phrase, a template of English numeral phrase, and a property item of representing the numeral sequence in Chinese-English translation. The variable of template of Chinese numeral phrase is the numeral itself. For each numeral variable, there are a property of its meaning and a property of the translation method. For each Chinese numeral phrase template, there would be exactly one English numeral phrase template corresponding to it.

When we translate, firstly we need to extract all numeral from the sentence and put them into the identifiable numeral depository. Then replace these numeral with the uniform variables. Secondly, search in the template depository, which stores all identifiable numeral phrase templates. Find out the templates most suitable for this sentence. Next, decide the translation method by the property of each variable, and translate them one by one by using the identifiable numeral depository. Then determine the sequence of each variable in the English template by using the sequence property. Last, compare to the English template, translate those Chinese numeral phrases into English.

With the experiment in Section 5, the result raises from 0.2882 to 0.3117, increased by 0.0235.

4 decoding

The decoding process consists of two steps: a, the phrase translations are generated for the input text, which is done before the searching begins. b, the search process takes place, through which phrase translation model, language model, distortion model are applied. Both steps will be described in detail.

4-1 Translation Options

A phrase translation table can be achieved through a bilingual corpus by the methods introduced in Section 2. Given an input text, all the phrase translations concerned can be applied by searching through the translation table, each applicable phrase translation for the source language phrase is a translation option [6]. Each translation option stores some information of the source phrase, the target phrase and phrase translation probability.

4-2 Searching Algorithm

The phrase-based decoder we developed employs a beam search similar to the one used by [6]. Considering the difference of expression habit between Chinese and English, many words must be complemented when translating Chinese sentence into English, such as a, an, the, of…Such word is difficult to extract. Its fertility is zero and corresponds to NULL in IBM Model 4. We call them F-zerowords. So after every new hypothesis expanded, F-zerowords can be added, which means, a NULL is added after the source phrase translated. Since perhaps not all words of the input sentence are necessary to be translated, we select the final hypothesis of the best translation in the last several stacks according to their scores when tracing back. This is different from [6]. Let’s describe it in detail.

The decoder starts with an initial hypothesis. There are two kinds of initial hypothesis: one is an empty hypothesis where no source phrase are translated and no target phrases are generated, the other is generating F-zerowords and corresponding to a NULL we supposed at the beginning of the input text.

New hypotheses are expanded from the currently existing hypotheses as follows: If the target phrase of the existing hypothesis is F-zeroword, an untranslated phrase and one of it’s translation options are selected. If it is not F-zeroword, there are two choices: one is expanding to a hypothesis which is achieved as described, the other is expanding to a hypothesis by selecting one of the F-zerowords as output, and corresponding to a NULL which added into the input text after the source phrase of the existing hypothesis.

The hypotheses are stored in different stacks. Each of them has a sequence number. The odd stack s_{2p-1} contains all hypotheses whose target phrases are not F-zerowords and in which p source words have been translated so far. (If the target phrase of the hypothesis is not F-zeroword, it stored in the stack 2p-1, p is the number of source words translated), the even stack s_{2p} contains all hypotheses whose target phrases are F-zerowords and in which p source words have been translated accumulatively. We recombine search hypotheses as described in [10], and prune out weak hypotheses based on the probability they incurred so far and a future score estimated as in [6]. All these reduce the number of hypotheses stored in stacks to speed up the decoder.

The current probability of the new hypothesis is the probability of the original hypothesis multiplied with the translation, distortion and language probability of the added phrasal translation, the probability formula is:

$$p(e|c) = p_T(c|e)^{\lambda_t} \times p_L(e)^{\lambda_l} \times p_D(e,c)^{\lambda_d} \text{ (3-1)}$$

In which $p_T(c|e)$ is the translation model computed according to (2-3), $p_L(e)$ is the target language model in which a 3-gram (trigram) language model is applied. $p_D(e,c)$ is the distortion model which allows for reordering of the input sentence, it is computed as follows:

$$p_D(e,c) = \lambda |a_i - b_{i-1} - 1| \text{ (3-2)}$$

where a_i denotes the start position of the source phrase that was translated into the i th target phrase, and b_{i-1} denotes the end position of the source phrase that was translated into the $(i-1)$ th target phrase. Each model is weighted by a parameter. We take the λ value here as 1 temporarily. We can also take output sentence length model into account.

The hypotheses are generated continuously until all the words of the input sentence are translated. However, considering there are many auxiliary words and mood words in Chinese, and these words have no corresponding English words, we don’t require all words in source language to be translated. Supposing the length of source language sentence is L, we take a ratio ‘a’ according to experience. Then select the best sentence as the translation result from all candidate sentences longer than L*a.

$$S_{best} = \arg\max_s \{P_s\}$$

where P_s is the accumulative probability of the hypothesis S , The method we used (denoted as back1) is different from that in [6](denoted as back2). And the experiments show our method has better performance in Section 4.

5 Experiments

We carried a number of experiments on Chinese-to-English translation tasks. A 31.6M bilingual corpus is used as training data for comparing different phrase translation extraction methods, investigating the effect of F-zerowords and the trace back method we used . We used a 60.9M bilingual corpus as training data to test the different effect of some maximum numbers of translation options for each source phrase. 1000 sentences of length 5-20 were reserved for testing of all the experiments.

5.1 Comparison of Different Phrase Translation Extraction Approaches

First, we compared the performance of the four methods and their combination for phrase translation extraction: extracting phrase pairs directly through IBM Model 4 (EDM), from HMM alignment model (HMM), integrated segmentation and phrase alignment (ISA) and Giza++ toolkit (Giza++). Table 1 shows the results of each method and their combination. All experiments used the decoder we described in Section 3.

Table 1

Method	Training corpus size	Size of phrase pair extracted	Bleu (4-gram)
EDM	31.6M	194802 pairs	0.2683
ISA	31.6M	187011 pairs	0.2751
HMM	31.6M	278770 pairs	0.2637
Giza++	31.6M	695486 pairs	0.2882
Combing methods above	31.6M	1077049 pairs	0.2887

From table 1, we see that each phrase translation extraction approach gives different phrase pair numbers and translation results. The phrase pairs number from ISA is the smallest, EDM only extracts phrase pairs whose source language phrase is composed of two or three words, but the translation results of EDM and ISA are almost the same, the HMM a little inferior to them. The Giza++ extracts the most phrase pairs of the four methods, the translation result from it is superior to other methods. Combing these methods always leads to a little better result.

5.2 Comparison of back2 and back1

We also performed experiments to compare back1 and back2, the results are shown in table2. In the table, M means word-based translation model, +NF0 means F-zerowords are not applied, +F0 means F-zerowords are applied, +NUM means number translation. We can see the result of the word-based system with no F-zerowords and BACK2 is the lowest . When the tracing back method used in [6] is replaced by the method proposed by us, the result rises (increases) 0.0086 from 0.1833 to 0.1919 with no F-zerowords. The result increases more obviously form 0.2372 with back2 to 0.2663 when F-zerowords are added. When extracting phrase by Giza++, the result also goes up owning to using back1. All these show back1 is superior to back2 because some source language words are not necessary to be translated .

Table 2

Method	Training corpus size	Bleu (4-gram)
M+NF0+BACK2	31.6M	0.1833
M+NF0+BACK1	31.6M	0.1919
M+F0+BACK2	31.6M	0.2372
M+F0+BACK1	31.6M	0.2663
(Giza++)+NF0+BACK2	31.6M	0.2730
(Giza++)+NF0+BACK1	31.6M	0.2864
(Giza++)+F0+BACK2	31.6M	0.2763
(Giza++)+F0+BACK1	31.6M	0.2882
EDM+NF0+BACK1	31.6M	0.1978
EDM+F0+BACK1	31.6M	0.2683
(Giza++)+F0+BACK1+NUM	31.6M	0.3117

5.3 the role of F-zerowords

From table 2, when F-zerowords are added through the decoding of word-based system, the result goes up sharply from 0.1919 to 0.2663 with back1, increasing by 0.0744, which denotes F-zerowords play a important role. This is because some words such as art. , prep. are complemented under the drive of language model, distortion model, which makes the output sentence more reasonable. The same conclusion can be drawn when phrased extracted directly. But when we use Giza++ to extract phrase pair, the results almost remain the same when the same trace back method is used, which is because with the phrase number rising, some F-zerowords are extracted in the phrase, and the effect of F-zerowords is minified.

5.4 the number of translation options for each source phrase

A strategy to limit the search space is reducing the number of translation options for each source phrase, we experiment on a 60.9M corpus, the results are shown in Table 3.

Table 3

Methods	Bleu (4-gram)	Decoding time
G+F0+back1	0.3418	2H6M
G+F0+back1_sort100	0.3452	40M
G+F0+back1_sort150	0.3446	54M
G+F0+back1_sort200	0.3423	64M
G+F0+back1_sort50	0.3366	23M

In Table 3, _sortn means selecting n translation options of the highest probability for each source phrase, 100 translation options (_sort100) proved to be sufficient. When translating 1000 sentences of 5-20 words, the result increases from 0.3418 to 0.3452, and the decoding time drops form 126 minutes to 45 minutes. Obviously we achieved fast decoding and better performance.

5.5 C_star Test Result

The test result of C_star in 2005 is shown in Table 4. ASR is the result after speech recognise. We just selected the first of 20 candidates of speech recognise result to translate. Manual transcript is the result of directly text translation. Because we need to translate numeral phrase first, we didn't use the result of given document. We combine them, seperate and mark them, then handle the result.

Table 4

	ASR Output	manual transcrip
Training corpus size	1.5M bilingual sentences	1.5M bilingual sentences
BLEU	0.3845	0.5279
NIST	8.0406	10.2499

6 Conclusion

In summary, this paper presents a phrase-based statistical machine translation system including methods to extract phrase translations from a bilingual corpus, the phrase translation model, along with the decoding framework. Our experiments show that phrase-based translation gets much better performance than traditional word-based methods. The F-zerowords usually play an important role in decoding, and the tracing back method we used is superior to that used in [6]. Selecting a certain number of top-high-probability translation options for each source phrase may lead to fast decoding speed and high quality.

Although we apply four methods to extract phrase pairs, for some source language phrase, the better translation option’s probability is not ensured to be higher than that of bad ones. We plan to do some studies about processing the phrase pairs extracted and computing the phrase translation’s probability.

7 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, pp. 263-311,1993.

[2] Yeyi Wang and Alex Waibel. Fast Decoding for Statistical Machine Translation. Proc. ICSLP 98, Vol. 6,pp.2775-2778,1998

[3] F. J. Och and H. Ney. Improved Statistical Alignment Model. Proceeding of ACL-00,PP. 440-447,2000.

[4]Yamada, K. and Knight. A Syntax-based Statistical Translation Model. In Proc. of the 39th Annual Meeting of ACL, 2001

[5] March, D. and Wong W. A Phrased-Based, Joint Probability Model for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP.2002.

[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] Ying Zhang, Stephan Vogel and Alex Waibel. Integrated Phrase Segmentation and Alignment Model for Statistical Machine Translation. Submitted to Proc. of International Conference on Natural Language Processing and Knowledge Engineering(NLP-KE), 2003.

[8] Stephan Vogel, Hermann Ney, and Christoph Tillmann . HMM-based Word Alignment in Statistical Translation. in COLING'96: The 16th Int. Conf. On Computational Linguistics,pp.836-841, 1996.

[9] Och, F. J., Tillmann, C., and Ney, H. improved alignment models for statistical machine translation. In proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora, pages 20-28,1999.

[10] Och, F. J., Ueffing, N., and Ney, H. An efficient A^* search A^* algorithm for statistical machine translation. In Data-Driven MT Workshop. 2001.