A Hybrid Approach for Spoken Language Machine Translation

Wenhan Chao[1] Zhoujun Li[2] Yuexin Chen[1]

[1]School of Computer Science, National University of Defense Technology, Changsha 410073, P.R. China
[2]School of Computer Science and Engineering, Beihang University, Beijing 100083, P.R. China

Abstract

In this paper, we propose a hybrid approach, which is a statistical machine translation (SMT), while using an example-based decoder. In this way, it will solve efficiently the re-ordering problem in SMT and the problems for spoken language MT, such as lots of omissions, idioms etc. We present a novel re-ordering model for SMT firstly and then an example-based decoder. Through experiments, we show that this approach obtains significant improvements over the baseline on a Chinese-English spoken language translation task.

Keywords: SMT, EBMT, Re-ordering model

1. Introduction

The are-of-the-state statistical machine translation (SMT) model is the log-linear model [1]:

$$\Pr(E \mid C) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(E,C)]}{\sum_{E'} \exp[\sum_{m=1}^{M} \lambda_m h_m(E',C)]} \quad (1)$$

where $h_m(E,C)$ represents the features and λ_m is the weight of the feature $h_m(E,C)$.

This log-linear model provides a framework, which can be used to incorporate any useful knowledge for machine translation, such as translation model, language model etc.

Considering the difference in word order between two languages, the word and phrase reordering is very important in the SMT systems. In the current phrase-based SMT systems, they handle the localized word reordering through phrases, and then using the phrase reordering model and language model to restrict the order of the phrases. The phrase re-ordering model may be simple or complex, due to the relationship in the word order between the two languages. For two languages which are close, such as French and English, the simple distortion model [2] may achieve good results, which only considers the jump distance between the target phrases for adjacent source phrases. However, for two languages which are very different in word order, such as Chinese and English, the distortion model is not enough.

Many researchers have proposed different reordering models. Some of them [3]-[4] predicate the reordering of the adjacent phrase pairs, and they only handle local reordering. And the others [5]-[9] handle the global reordering, i.e., they predicate the reordering of long distances. In the global models, Nagata's [5] clustered model predicates the reordering based on the current phrase pair and the previous phrase pair; Yamata [6] restricts the phrases order by the syntax tree, and Wu [6] proposes an Inversion Transduction Grammar (ITG), Chiang [8] presents a hierarchical model which both use a *formally* syntax-based model to constrain the order of the phrases. Xiong's [9] maximum entropy model is derived from the ITG model, which transforms the problem predicating the reordering into a classification problem.

For the spoken language machine translation, there are some other problems deriving from the characteristics of the spoken language. In a spoken text, the phenomenon of omission is common, the order of the words may be more flexible than written text, and there may be many idioms which must be kept as a whole. These problems increase the complexity of the translation. However, there is also a useful feature for the spoken text; most of the spoken text is shorter than the written text.

In this paper, we propose a hybrid approach, which is a SMT system, while using an example-based decoder. In the system, we use a novel phrase reordering model, which is derived from bracketing ITG model and combines the advantages of the local reordering models. And we also present an example-based decoder which may incorporate the re-ordering model to improve the spoken language translation. The rest of this paper is organized as follows: Section 2 presents how to derive the new reordering model from the bracketing ITG and how to build the model from the bilingual corpus. In Section 3, we design the decoder. In Section 4, we test our model and compare it with the baseline system. Then, we conclude in Section 5 and Section 6.

2. Re-ordering model

Since our reordering model is based on the Bracketing ITG model, we will introduce the ITG model firstly, and then propose how to transform it to achieve our model.

2.1. Bracketing ITG model

Bracketing ITG is a synchronous context-free grammar, which generates two output streams simultaneously. It consists of the following five types of rules:

$$A \xrightarrow{a[]} [AA] \quad \textbf{(2)}$$

$$A \xrightarrow{a<>} < AA > \quad \textbf{(3)}$$

$$A \xrightarrow{bij} c_i / e_j \quad \textbf{(4)}$$

$$A \xrightarrow{bi\varepsilon} c_i / \varepsilon \quad \textbf{(5)}$$

$$A \xrightarrow{b\varepsilon j} \varepsilon / e_j \quad \textbf{(6)}$$

where A is the only non-terminal symbol, [] and <> represent the two operations which generate outputs in straight and inverted orientation respectively. c_i and e_j are terminal symbols, which represent the words in both languages, ε is the null words. The $a_{[]}$, $a_{<>}$, b_{ij}, $b_{i\varepsilon}$ and $b_{\varepsilon j}$ are the probabilities of the rules. The last three rules are called lexical rules.

In this paper, we consider the phrase-based SMT, so the c_i and e_j represent phrases in both languages, which are consecutive words, and the number of words within the phrase is called the length of the phrase. And a pair of c_i and e_j is called a phrase-pair, or a block. In rules (2) and (3), the A in the left side is composed of the two As in the right side, so we call the left A parent block, and the right A child block. The block generated from lexical rules is called an atom block.

During the process of decoding, each phrase c_i in the source sentence is translated into a target phrase e_j through lexical rules, and then rules (2) or (3) are used to merge two adjacent blocks into a large block in straight or inverted orientation, until the whole source sentence is covered. In this way, we will obtain a binary branching tree, which is different from the traditional syntactical tree, and in which each constituent is a block, which only needs to satisfy the consecutive constraint.

Since the ITG model only needs to preserve the constituent structure, it achieves a great flexibility to interpret almost arbitrary reordering during the decoding, while keeping a weak but effective reordering constraint in the global scope. Figure 1 gives an example to illustrate a derivation from the ITG model.

On the other hand, the $a_{[]}$ and $a_{<>}$ in the rules (2) and (3) are independent of the blocks in the right side, they only represent the preference to choose straight or inverted orientation. Thus, it is hard to predict the local reordering of adjacent blocks.

In this paper, we hope to find an approach which will strengthen the model's ability to predict the local reordering while keep the global constraint.

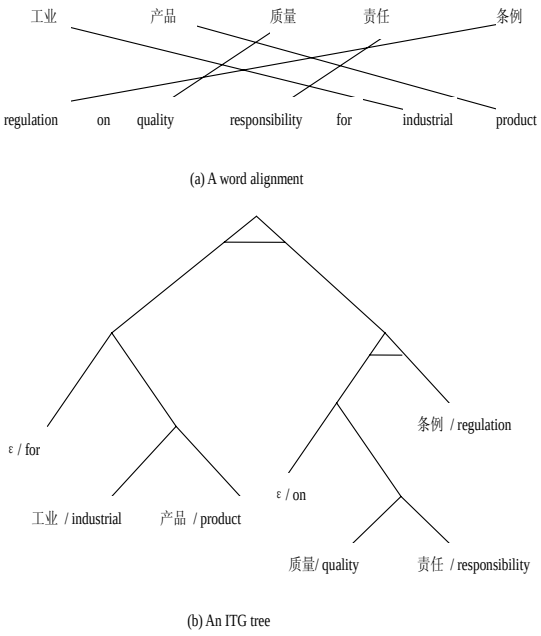

Fig. 1: An ITG tree which is formed from a word alignment. A line between the branches means an inverted orientation, otherwise a straight one.

2.2. New Reordering Model

Our problem is to predicate the reordering $o \in \{straight, inverted\}$ of any two adjacent blocks A^1 and A^2, we use $r(o, A^1, A^2)$ to represent it. A straightforward method is to compute the co-occurrence count between A^1 and A^2, and the frequencies they are in straight or inverted orientation respectively. And then use the MLE to predicate the reordering probabilities.

However, due to the constraints for corpus size and the memory of the computer, it is impossible to collect the reordering of any two blocks; and in

general, the larger the block is, the smaller the frequency it occurs, so that the reordering probabilities are not accurate.

So, instead of recording all the reordering of any two blocks, we predicate the reordering of each atom block A^o and any other block A^* which is preceding or posterior to the A^o. Generally, an atom block is shorter and the count it occurs will be larger, so the predication will be more accurate. For each atom block A^o, there are the following four reordering:

- $a^o_{[]*}$: The probability of $O(A^o, A^*) = straight$.
- $a^o_{<>*}$:the probability of $O(A^o, A^*) = inverted$.
- $a^o_{*[]}$: The probability of $O(A^*, A^o) = straight$.
- $a^o_{*<>}$: The probability of $O(A^*, A^o) = inverted$.

Now, deriving from the rules (2) and (3), we will be able to predicate the reordering of any two blocks A^1 and A^2.

- If A^1 and A^2 are both atom blocks, then

$$r(straight, A^1, A^2) = a^1_{[]*} \bullet a^2_{*[]} \quad (7)$$

$$r(inverted, A^1, A^2) = a^1_{<>*} \bullet a^2_{*<>} \quad (8)$$

  I.e., if the reordering of A^1 and A^2 is straight, the probability is the product of the $a^1_{[]*}$ and $a^2_{*[]}$, where $a^1_{[]*}$ is the probability that $O(A^1, A^*) = straight$ and $a^2_{*[]}$ is the probability that $O(A^*, A^2) = straight$. In the same way, if the reordering of A^1 and A^2 is inverted, the probability is the product of the $a^1_{<>*}$ and $a^2_{*<>}$.
- If A^1 or A^2 are not atom blocks, then we can always find the rightest child block A^1_o of A^1 which is an atom block, and the leftest child block A^2_o of A^2 which is an atom block. And then we use formula (6) and (7) to predicate the $r(o, A^1, A^2)$. That is, we use the two adjacent atom blocks within A^1 and A^2 respectively to represent the whole blocks.

In this way, we can predicate the $r(o, A^1, A^2)$ of any two blocks, which can be large or small, if only the blocks satisfy the constituent structure. During decoding, a sequence of application of rules (2) and (3) is the process of phrase reordering, so we define the reordering model independently as follows:

$$\Pr(O) = \prod_i r(o_i, A^{i1}, A^{i2}) \quad (9)$$

where $r(o_i, A^{i1}, A^{i2})$ is the probability to apply the rules (2) or (3) in the i-th time.

2.3. Building the re-ordering model

In order to train the translate model and the reordering model, we use a word-aligned bilingual corpus, in which the word alignments satisfy the ITG constraint. Figure 1 illustrates a valid word alignment example, which satisfies the ITG constraint, and forms a binary branching tree, in which the leaves represent the aligned word pair. In our word alignment, the each word pair may consist of multi words in both sentences, but they must be consecutive.

For the word alignment forms a hieratical binary tree, we can extract the blocks in a straight-forward way, i.e. choosing each constituent as a block. Due to the memory constraint, we restrict the maximum length N of each block. In this paper, we set the N = 5. Because we need compute the lexical translation model and reordering model, we collect the following information at the same time when collecting the each block A^o:

- The word alignment within the block A^o. For the word alignment is hieratical, i.e. each constituent may be a leaf or consist of two child constituents. We record the information whether the constituent is a leaf or the division of the child constituents.
- The reordering of the block A^o and the preceding or posterior blocks.

Thus, the final information of each block A^o consists of the block text, frequency of the block, lexical alignment, and the reordering. Table 1 lists the blocks which are extracted from the word alignment example in Figure 1. The number in reordering column represents respectively the frequency of the reordering of the block A^o and the preceding and posterior blocks.

After collecting all the blocks in each word alignment in the trained bilingual corpus, we combine the same blocks and obtain the final block table.

The block table contains the reordering information for each block, which consists of the frequencies in straight and inverted orientation. So, we can obtain the reordering model in the following way:

$$a^{ce}_{[]*} = \frac{\text{frequency of } (O(ce, A^*) = straight)}{\text{frequency of the block (c, e)}}$$

$$a^{ce}_{<>*} = \frac{\text{frequency of } (O(ce, A^*) = inverted)}{\text{frequency of the block (c, e)}}$$

$$a^{ce}_{*[]} = \frac{\text{frequency of } (O(A^*, ce) = straight)}{\text{frequency of the block (c, e)}}$$

$$a^{ce}_{*<>} = \frac{\text{frequency of } (O(A^*, ce) = inverted)}{\text{frequency of the block (c, e)}}$$

Through the block table, we can obtain the translate models $P(e|c)$, $P(c|e)$, $P_w(e|c)$, $P_w(c|e)$, by using the frequencies of the blocks, the first two models are phrase translation models, and the last two are lexical translation models.

Chinese	English	Freq	Align	Reordering
工业	industrial	1	1-1	1 0 1 0
产品	product	1	1-1	1 0 0 1
工业 产品	industrial product	1	1-1; 2-2	1 0 0 1
工业 产品	for industrial product	1	1-2; 2-3	1 0 0 1
质量	quality	1	1-1	0 1 1 0
责任	responsibility	1	1-1	1 0 0 1
质量 责任	quality Responsibility	1	1-1;2-2	0 1 0 1
质量 责任	on quality responsibility	1	1-2;2-3	0 1 0 1
条例	regulation	1	1-1	0 1 1 0
质量 责任 条例	regulation on quality responsibility	1	1-3; 2-4; 3-1	1 0 1 0

Table 1: The blocks extracted from word alignment in Fig1.

3. Example-based Decoder

The decoder searches the best E^* when given a source sentence C.

$$E^* = \arg\max_E \{\Pr(E|C)\} \quad (10)$$
$$= \arg\max_E \{\exp[\sum_{m=1}^{M} \lambda_m h_m(E, C)]\}$$

For our SMT system applies a reordering model which is based on the ITG model, so that the target sentence E^* should satisfy the ITG constraint, i.e., C and E form a hieratical binary branching tree.

Generally, we can implements a CKY style decoder, which using the re-ordering model to predicate the re-ordering of two adjacent blocks.

However, in the spoken language, the word order is very flexible, and there also exists a lot of omissions and idioms.

So, we wish to constrain the re-ordering of two blocks further, and have the chance to translate the omitted words. In this paper, we use an example-based decoder to solve these problems.

3.1. Retrieval of Examples

When given an input sentence C_0, the decoder firstly retrieves a collection of translation examples {(C_1, E_1, A_1) ,((C_2, E_2 , A_2),.... }, where A_k is the word alignment for each translation example, and C_k is similar with C_0 .

In order to obtain the similarity between C_k and C_0, a straight-forward method is to compute the edit distance, by giving each operation insertion, deletion and substitution a distance one.

Because the training corpus may be large, the complexity will be very large for each input. So, in our decoder, we will use an easy way.

In the above section, we have described that we have obtained a block table, and the translate models. Now, for each input sentence C_0, we collect the probable monolingual phrases, and then search the blocks in the translate models, and sort them by the input phrase, the probability $P(e|c)$. For the blocks with the same input phrase, we only keep the best N. In this paper, we set N = 5.

After collecting the probable blocks, we use them as patterns to match the examples. If there exists at least one pattern in a translation example, we take it as a valid example. For each block, if it has occurred at least M times in the valid examples, we remove it from the pattern set. (Here M = 5.) If the pattern set is NULL, the retrieving process stops.

In this way, we can retrieve the valid examples quickly. The following section presents how to use them to decode.

3.2. Decoding

After retrieving the translation examples, our goal is to use these examples to constrain the order of the output words. During the decoding, we iterate the following two steps:

- Matching

  For each translation example (C_k, E_k, A_k) consists of the word alignment, which satisfies the ITG constraint, i.e., forms a binary branching tree. So we can match the input sentence with the tree, and get some translation templates for each translation example, in which some input words (monolingual phrases) are translated and they must maintain the constituent structure constraint, and some phrases are un-translated, and they are taken as new inputs

(called child inputs) respectively, and match the translation examples iterately.

Note when matching an input with each example, we may get one or more templates. For example, if there exists null-aligned output words which are adjacent to the matched blocks, then we may generate more templates to include them.

- Merging

If one child input is translated wholly, i.e. no phrase is un-translated. Then, it should be merged into the parent translation template. When merging, we must satisfying the ITG constraint, so we use the rules (2) and (3) to merge the child input with the adjacent blocks. The orientation is predicated by the re-ordering model. In fact, we can try all merging choices, including the orientation, and the blocks with which to merge, and then compute the scores of all these translation candidates.

In the end, we will obtain a collection of the translation candidates, the decoder sorts them by their scores, and output the best candidate. Figure 2 illustrates the decoding process.

4. Experiments

We carried out experiments on a Chinese-English bilingual spoken language corpus, and compared with the state-of-the-art distortion-based decoder Moses [10], which is an extension to the Pharaoh [11]. Table 2 shows the statistics of the training corpus, development corpus and test corpus.

For the baseline system, we used the default features: language model, translate models which were similar to ours, distortion model, word penalty and phrase penalty. We ran the default trainer in the Moses to train all of models and tune the feature weights, in which the trainer used the Giza++ [12] to achieve word alignment and minimum error rate to tune the weights. And then we run the decoder in the Moses on the test set.

For our SMT model, after obtaining the word alignment for each sentence pair in the corpus, which satisfied the ITG constraint, we collected the atom blocks and built the block table. And then we trained the translation models and reordering model, and tuned the feature weights. We tested our SMT system using two decoders, one is a CKY style decoder, and the other is the example-based decoder in section 3.

The results are listed in Table 3. The first line is the result of the baseline system, the second and third line is the results of our system which using the CKY style decoder and applying the reordering model or not respectively. The fourth and fifth are the results of our system which using the example-based decoder and applying the reordering model or not.

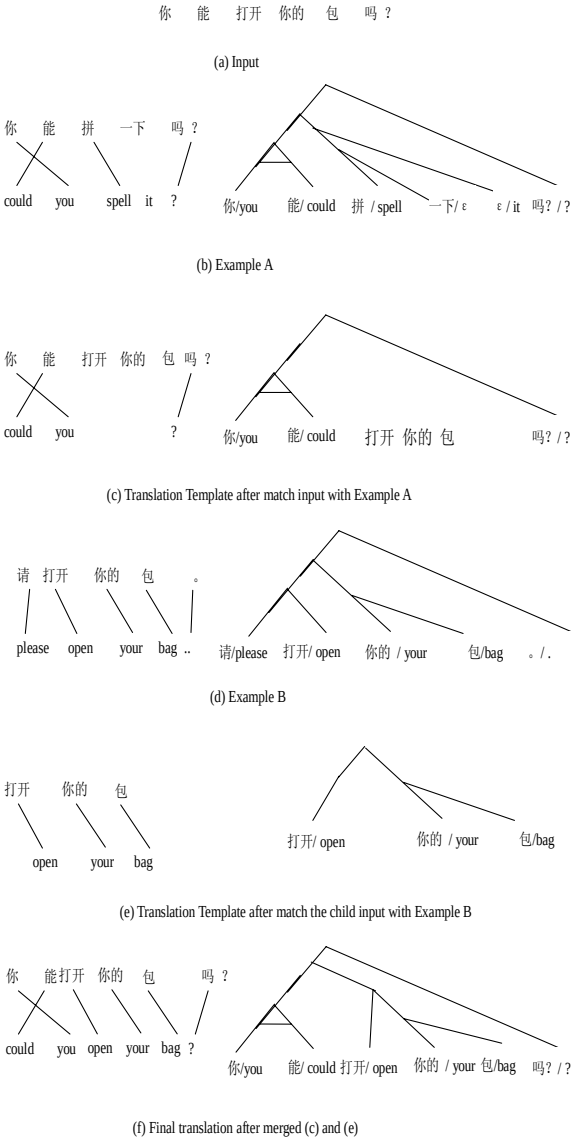

Fig. 2: An example to illustrate the example-based decoding process, in which there are two translation examples.

From the Table 3, we observed that the results of the baseline system and our CKY-reorder are close for they applied the similar features except that we obtained the block table in a different way. While the CKY+reorder achieves a significant improvement (about 5%) over the baseline system.

In addition, our EB-reorder obtains close result with the CKY+reorder system, and EB+reorder achieves an improvement (about 3%) over the CKY+reorder.

We concluded that our reordering model and the example-base decoder are both useful to solve the re-

ordering problem. And the example-based decoder is effective for the spoken language translation.

		Chinese	English
Training Corpus	Sentences	51,211	51,211
	Words	543,241	559,522
	Vocabulary	17,624	17,873
Develop Corpus	Sentences	500	500
	Words	5,837	5,912
Test Corpus	Sentences	500	500
	Words	5,337	5,452

Table 2: The statistic of the corpus.

System	Bleu (%)
Moses	33.12
CKY-reorder	33.65
CKY + reorder	34.93
EB-reorder	34.63
EB+reorder	35.87

Table 3 Results on baseline system and our system.

5. Related works

There are lots of works on the phrase re-ordering model, which may be divided into two categories, one is local model which predicates the local reordering of the adjacent blocks, and the other is global model, which can predicate the reordering of long distances.

Our re-ordering model is derived from the bracketing ITG, which is a global model, but is block independent, i.e., it only considers the preference of the output orientation. However, our model is block dependent, which predicates the re-ordering for the concrete two adjacent blocks.

Xiong's [9] maximum entropy model is derived also from the ITG model, and when predicating the re-ordering, it extracts the features in the two adjacent blocks and make a classification using the trained feature weights.

There is also some works about the hybrid machine translation. Watanabe [13] presents an example-based decoder for SMT, which using an information retrieval framework to retrieve the translation examples, and when decoding, which runs a hill-climbing algorithm to modify the translation example (C_0, E_k, A_k) to obtain an alignment (C_0, E'_k, A'_k).

6. Conclusion

In this paper, we proposed an ITG-based reordering model, which can integrate the local and global model. Our model explicitly models the reordering of each atom block A^o and any blocks which are preceding or posterior to A^o, and then through satisfying the ITG constraint, we may predicate the reordering of any two blocks which may be small or large.

We also presented an example-based decoder, which could incorporate the above re-ordering. Experiments on a Chinese-English bilingual spoken language corpus showed that our hybrid approach achieves an improvement over the baseline system.

In the future, we will test our system on the written text corpus, and we also need more effective methods to retrieve the translation examples when given larger training corpus.

References

[1] F.J. Och, H. Ney, Discriminative training and maximum entropy models for statistical machine translation. *Proceedings of the 40th Annual Meeting of the ACL*, pp. 295–302, 2002.

[2] P. F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer, The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2): 263–312, 1993.

[3] C. Tillmann, T. Zhang, A localized prediction model for statistical machine translation. *Proceedings of the 43rd Annual Meeting of the ACL*, pp. 557-564, 2005.

[4] S. Kumar, W. Byrne, Local phrase reordering models for statistical machine translation. *Proceedings of Human Language Technology Conference and Conference on Empirial Methods in Natural Language Processing (HLT/EMNLP)*, pp. 161-168, 2005.

[5] M. Nagata, K. Saito, A clustered global phrase reordering model for statistical machine translation. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 713-720, 2006.

[6] K. Yamada, K. Knight, A syntax-based statistical translation model. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 523–530, 2001.

[7] D. Wu, Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3): 377-404, 1997.

[8] D. Chiang, A hierarchical phrase-based model for statistical machine translation. *Proc. of ACL 2005*, pp. 263–270, 2005.

[9] D. Xiong, Q. Liu, S. Lin, Maximum entropy based phrase reordering model for statistical machine translation. *Proceedings of the 21st International Conference on Computational*

Linguistics and 44th Annual Meeting of the ACL, pp.521-528, 2006.
[10] http://www.statmt.org/moses/.
[11] P. Koehn, Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pp. 115-124, 2004.
[12] F.J. Och, H. Ney, A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52, March, 2003.
[13] T. Watanabe, E. Sumita, Example-based decoding for statistical machine translation. *Machine Translation Summit IX*, pp. 410-417, 2003.