NUDT Machine Translation System for IWSLT2007

Wen-Han Chao[1], *Zhou-Jun Li*[2]

[1]School of Computer Science, National University of Defense Technology, China
cwh2k@163.com
[2]School of Computer Science and Engineering, Beihang University, China
lizj@buaa.edu.cn

Abstract

In this paper, we describe our machine translation system which was used for the Chinese-to-English task in the IWSLT2007 evaluation campaign. The system is a statistical machine translation (SMT) system, while containing an example-based decoder. In this way, it will help to solve the re-ordering problem and other problems for spoken language MT, such as lots of omissions, idioms etc. We report the results of the system for the provided evaluation sets.

1. Introduction

The state-of-the-art statistical machine translation (SMT) model [1][2] is the log-linear model [3], which provides a framework to incorporate any useful knowledge for machine translation, such as translation model, language model etc.

In a SMT system, one important problem is the re-ordering between words and phrases, especially when the source language and target language are very different in word order, such as Chinese and English.

For the spoken language translation, the re-ordering problem will be more crucial, since the spoken language is more flexible in word order. In addition, lots of omissions and idioms make the translation more difficult.

In this paper, we present our hybrid translation system, which is a SMT system, while using an example-based decoder, which will use the translation examples to keep the translation structure, i.e. constraint the reordering, and make the omitted words having the chance to be translated.

In our system, each translation example is a triple (*C*, *E*, *TA*), where *C* represents the Chinese sentence, *E* the English sentence, and *TA* is the word alignment between *C* and *E*, which satisfies the inversion transduction grammar (ITG) [4] constraint, i.e. the *TA* forms a constituent structure tree.

This paper is organized as follows. In Section 2, we describe the various components in our system, especially the word aligner and decoder. In section 3, we report the experimental results of Chinese-English translation, and we conclude in section 4 and provide avenues for further research.

2. System Description

Our machine translation system is a modular MT engine, which mainly consists of the following components:

- *Word Alignment*: taking the bilingual sentence-aligned training corpus as input, obtains the Viterbi word alignment for each sentence pair, in our system, the word alignment must satisfy the ITG constraint.
- *Phrase Pair Extracting*: taking the bilingual word-aligned training corpus as input, extracts the valid phrase pairs and builds the translation model and the reordering model.
- *Decoder*: given a Chinese sentence as input, search the best translation using the word-aligned corpus and the translation model, reordering model and language model.

2.1. Word Alignment

The word alignment [5] is the base of the SMT system. In our system, the word alignment for each sentence pair is used to build translation model and reordering model, and also used to provide the valid translation example.

In our system, the word alignment needs to satisfy the ITG constraint, which is derived from the ITG grammar. The ITG is a synchronous PCFG, consisting of five types of rules:

$$A \longrightarrow [AA] \,|< AA >|\, c_i / e_j \,|\, c_i / \varepsilon \,|\, \varepsilon / e_j \qquad (1)$$

Where *A* is the 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. And each rule will be assigned a corresponding probability. The last three rules are called lexical rules.

A word alignment statisfying the ITG will form a binary branching tree, see Figure 1. And it provides a flexible but effective way to interpret almost arbitrary word order.

Wu[4] provides a DP algorithm to obtain the word alignment which satisfies the ITG constraint, we [6] have transferred the constraint to four simple position judgment procedures in an explicit way. So we can incorporate the ITG constraint as a feature into a log-linear word alignment model [7].

Given a sentence pair (*C*,*E*) , a log-linear word alignment is to find the best $A_{\max}$, so that:

$$A_{\max} = \arg\max_{A} \sum_{i=1}^{n} \lambda_i f_i(C, TA, E) \qquad (2)$$

Where the f_i represents the feature and λ_i is the corresponding weight of the feature.

In our word alignment model, it consists mainly of the following three features:

- *ITG constraint*: counts the number of links in the word alignment, which violating the ITG constraint. In order to ensure that the result word alignment satisfies the constituent structure, we set a very small negative weight for this feature, so that the word alignment will not be used whenever this feature occurs.

- *Conditional Probability Model:* we use a conditional probability as our base feature which accounts for the word correlation,

$$f_p(C,TA,E) = \log P(A|C,E) = \sum \log p(a|c,e) \qquad (3)$$

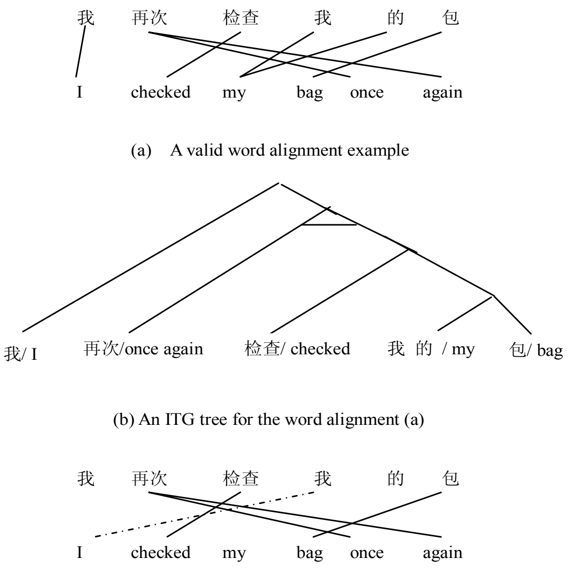
  Where $p(a|c,e)$ is the alignment probability when c and e co-occur.

- *Distortion Model:* we count the jump distance for this model:

$$f_d(C, TA, E) = \sum_i d_i \qquad (4)$$

  where the d_i represents the jump distance for each link in the word alignment, using one of the sentences as a reference.

We [6] use a beam search algorithm to find the Viterbi word alignment which is similar with the competitive linking algorithm [8]. And we tune the feature weights using the perception training [7], over a development set we aligned manually. In the end, we will obtain the bilingual word-aligned training corpus, in which each word alignment satisfies the ITG constraint, i.e., it forms a constituent structure tree.



(a) A valid word alignment example

(b) An ITG tree for the word alignment (a)

(c) An invalid word alignment example

Figure 1: *A valid word alignment (a) and the corresponding ITG tree (b) where the line between the branches means an inverted orientation, otherwise a straight one, and an invalid alignment example, where the dot line is an invalid link when given the other links.*

2.2. Phrase Pair Extracting

In our SMT model, we use the translation models and the re-ordering model as features. And we use a word-aligned bilingual corpus, in which each word alignment satisfies the ITG constraint.

For the word alignment forms a hierarchical binary tree, we can extract the phrase pairs in a straight-forward way, i.e. choosing each constituent as a phrase pair, called a block. We can also collect the reordering information between two blocks according to the orientation of the branches.

Thus, we will build the translation models $P(e|c)$, $P(c|e)$, $P_w(e|c)$, $P_w(c|e)$, using the frequencies of the blocks, and the re-ordering model $P(o|b_1,b_2)$ in the following way:

$$p(straight \mid b_1,b_2) = \frac{\text{freq. of } (O(b_1,b_2) = straight)}{\text{freq. of } cooccur\,(b_1,b_2)} \qquad (5)$$

$$p(invert \mid b_1,b_2) = \frac{\text{freq. of } (O(b_1,b_2) = invert)}{\text{freq. of } cooccur\,(b_1,b_2)} \qquad (6)$$

2.3. Decoder

2.3.1. A baseline decoder

In order to satisfy the ITG constraint, we regard the process of the decoding as a sequence of applications of rules in (1), i.e., the (*C*,*E*) will be a derivation *D* of the ITG. Following Och and Ney[4], we define the probability for each rule as:

$$\Pr(rule) = \prod_i h_i(rule)^{\lambda_i} \qquad (7)$$

Where the h_i represents the feature and λ_i is the corresponding weight of the feature.

- If the rule is a lexical rule, then we will consider the four translation models in the section 2.2, i.e. the $P(e|c)$, $P(c|e)$, $P_w(e|c)$, $P_w(c|e)$, as features.
- Otherwise, we will only consider the re-ordering model in Section 2.2.

And the probability for the derivation will be:

$$\Pr(D) = \prod_{r \in D} \Pr(r) \bullet \Pr_{lm}(E)^{\lambda_{lm}} \qquad (8)$$

Where the $\Pr_{lm}(E)$ is the language model. So the decoder searches the best *E** derived from the best derivation *D**, when given a source sentence *C*.

$$D^* = \underset{c(D)=C}{\arg\max} \Pr(D) \qquad (9)$$

In order to evaluate the example-based decoder, we develop a CKY style decoder as a baseline MT system, so that the (*E*,*C*) satisfies the ITG constraint.

2.3.2. The example-based decoder

The example-based [9] decoder consists of two components:

- *Retrieval of examples*: given the input Chinese sentence C_0 and the bilingual word-aligned corpus, collects a set of translation examples {(C_1, E_1, TA_1) ,((C_2, E_2 , TA_2),....} from the corpus, where the C_k in each translation example is similar to the input sentence.

- *Decoding*: given the input and the translation examples and the translation models, language models and re-ordering model, searches the best translation for the input.

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 easier way.

We collect the probable monolingual source phrases, which are consective words, in the input C_0 firstly. And for each source phrase, we search the phrase pairs in the translation model $\Pr(e \mid c)$ with the same source phrase, and sort them by the probability. For each source phrase, we only keep the best N phrase pairs (here $N = 10$).

After collecting the phrase pairs, 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 phrase pair, if it has occurred at least M times in the valid examples, we remove it from the pattern set. If the pattern set is NULL, the retrieving process stops. Thus, we can retrieve the valid examples quickly.

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 (Figure 2 shows an example).

- *Matching*

For each translation example (C_k, E_k, TA_k) consists of the constituent structure 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, and some phrases are un-translated. I.e., the template is a partial translation.

We call the un-translated phrases as child inputs, and try to translate them iterately, i.e., decoding them using the translation examples.

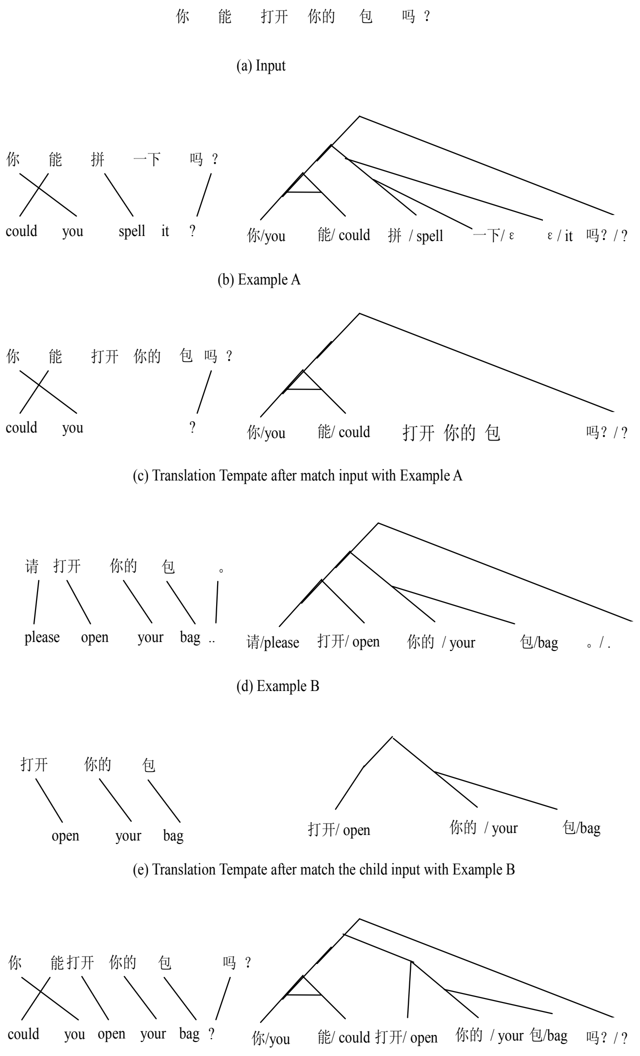
- *Merging*

If one child input is translated wholly, i.e. no phrase is un-translated. Then, it should be merged into the parent translation template to form a new template. If all child inputs are translated, then returning the final translation. When merging, we must satisfy the ITG constraint.

When decoding, we need to evaluate the translate template using the following function:

$$f(temp) = \log P(E_{trans} \mid C_{trans}) + \log H(C_{untrans}) \qquad (7)$$

Where $P(E_{trans} \mid C_{trans})$ is the probability for the translated phrases, which can be calculated using the SMT model, and the $H(C_{untrans})$ is the estimated score for the un-translated phrases which can also be estimated using the SMT.


你 能 打开 你的 包 吗 ?
你 能 拼 一下 吗 ?
could you spell it ?
你/you 能/ could 拼 / spell 一下/ ε ε / it 吗? / ?
你 能 打开 你的 包 吗 ?
could you ?
你/you 能/ could 打开 你的 包 吗? / ?
请 打开 你的 包 。
please open your bag ..
请/please 打开/ open 你的 / your 包/bag 。/ .
打开 你的 包
open your bag
打开/ open 你的 / your 包/bag
你 能打开 你的 包 吗 ?
could you open your bag ?
你/you 能/ could 打开/ open 你的 / your 包/bag 吗? / ?

(a) Input

(b) Example A

(c) Translation Tempate after match input with Example A

(d) Example B

(e) Translation Tempate after match the child input with Example B

(f) Final translation after merged (c) and (e)

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

3. Experiments

We carried out experiments on the Chinese-to-English translation task, which provides a sentence-aligned training corpus consisting of 39,953 Chinese-English sentence pairs, five development sets and one test set which consists of 489 Chinese sentences. We take the third development set, i.e. the *IWSLT07_devset3_**, to tune the feature weights.

In the training corpus, the Chinese sentences have been segmented, while the English sentences have not been tokenized. So, in the pre-processing step, we tokenized the English sentences using the tools from the WMT07 share task. And we also obtained the lowercase words for all English sentences.

Considering the size of the training corpus is relatively small, and the words in Chinese have no morphological changes, we stemmed the words in the English sentences by using a morphological dictionary, where each entry consists of one word and its stem word. Table 1 shows the statistics for the training corpus, development set and test set.

Table 1: *The statistics of the corpus*

		Chinese	English (stemmed)
Train. corpus	Sentences	39,963	
	Words	351,060	377,890
	Vocabulary	11,302	7,610
Dev. Set	Sentences	506	
	Words	3,826	
Test Set	Sentences	489	
	Words	3,189	

Because of the metrics of the evaluation campaign of IWSLT2007 take case information into account, in the post-processing step, we use two simple rules to obtain the case sensitive outputs, the first rule is the capital letter of the first word in each sentence must be uppercase, and the second one is the word "i" must be "I".

Firstly, we tested our machine system with all English sentences in both training corpus and the reference set are tokenized, low cased and stemmed. The results are showed in Table 2.

Table 2: *Test results with English sentences are stemmed*

Decoder	Bleu
CKY-Decoder	0.2741
EB-Decoder	0.3012

The first column lists the two decoders in our SMT system, and the second column lists the Bleu scores [10] for the two decoders. The results show that the example-based decoder achieves an improvement over the baseline decoder.

Secondly, we considered the case information, i.e. we used the two rules to post-process the output.

Also, we took into account the morphological changes of the English words. In order to find the most likely sequence, we use a 3-gram language model trained on an un-stemmed text. The 3-gram language model was trained on the English sentences of the training data, using the SRILM toolkit [11]. Table 3 lists the results.

Table 3: *Test results with English sentences are normal*

Decoder	Bleu
CKY-Decoder	0.1758
EB-Decoder	0.1934

The results show that the Bleu scores decrease quickly from the Table 2. We conclude that our method to handle the morphological changes is too easy.

4. Conclusions

In this paper, we proposed SMT system with an example-based decoder, which is derived from the ITG, for the spoken language machine translation. This approach will take advantage of the constituent tree within the translation examples to constrain the flexible word re-ordering in the spoken language, and it will also make the omitted words have the chance to be translated. Combining with the re-ordering model and the translation models in the SMT, the example-based decoder obtains an improvement over the baseline phrase-based SMT system.

In the future, we need more effective methods to retrieve the translation examples, and we also plan to improve the decoding model in the example-based decoder. In addition, we will improve the methods to handle the morphological changes from the stemmed English words.

5. References

[1] Philipp Koehn, Franz Josef Och and Daniel Marcu: "Statistical Phrase-Based Translation". *In NAACL/HLT 2003*, pages 127-133(2003)

[2] David Chiang: "A Hierarchical Phrase-Based Model for Statistical Machine Translation". *In Proc. of ACL 2005,* pages 263–270 (2005)

[3] Franz Joseph Och and Hermann Ney: "Discriminative training and maximum entropy models for statistical machine translation". *In Proceedings of the 40th Annual Meeting of the ACL*, pp. 295–302(2002)

[4] Dekai Wu: "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora". *Computational Linguistics*, 23(3):374(1997)

[5] Franz Joseph Och and Hermann Ney: "A Systematic Comparison of Various Statistical Alignment Models". *Computational Linguistics,* 29(1):19–52, March(2003)

[6] Wen-Han Chao and Zhou-Jun Li: "Incorporating Constituent Structure Constraint into Discriminative Word Alignment", *MT Summit XI, Copenhagen, Denmark, September 10-14, 2007,* accepted**.** (2007)

[7] R. Moore. "A discriminative framework for bilingual word alignment". *In Proceedings of HLT-EMNLP*, pages 81–88, Vancouver, Canada, October. (2005)

[8] I. Dan Melamed. "Models of Translational Equivalence among Words". *Computational Linguistics, 26(2)*:221–249. (2000)

[9] Taro Watanabe and Eiichiro Sumita: "Example-based Decoding for Statistical Machine Translation". *In Machine Translation* Summit IX pp. 410-417 (2003)

[10] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu: "BLEU: a Method for Automatic Evaluation of Machine Translation". *In Proceedings of the 40th Annual Meeting of the Association fo Computational Linguistics(ACL), Philadelphia, July 2002,* pp. 311-318(2002)

[11] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, 2002, pp. 901–904