Similar Sentence Retrieval for Machine Translation Based on Word-Aligned Bilingual Corpus

Wen-Han Chao[1] and Zhou-Jun Li[2]

[1] School of Computer Science, National University of Defense Technology, China, 410073
cwh2k@163.com
[2] School of Computer Science and Engineering, Beihang University, China, 100083
lizj@buaa.edu.cn

Abstract. In this paper, we present a novel method to retrieve the similar examples from the corpus when given an input which should be translated, in which we use the word alignment between the bilingual sentence pair to measure the similarity of the input and the example.

1 Introduction

Measures of the sentence similarity are very important in many NLP applications, such as machine translation (MT), especially the example-based machine translation (EBMT) [1], information retrieval (IR) and text summarization etc.

Recently, some researchers present a hybrid corpus-based machine translation system[2], which is a statistical machine translation (SMT), while using an example-based decoder, in which the similar translation examples will avoid translating from the scratch, and the similar examples retrieved will affect the results greatly.

When retrieving the similar examples, most of the metrics measure the similarity of the two monolingual sentences. However, in a hybrid machine translation system, the examples in the training corpus provide rich information, such as word alignment.

In this paper, we provide a novel approach, which use the bilingual information of the examples to retrieve the similar examples for an input sentence.

2 The Word-Aligned Bilingual Corpus for MT

There are two types of corpus-based MT systems: Statistical Machine Translation (SMT) and EBMT. SMT obtains the translation models during training, and does not need the training corpus when decoding; while EBMT retrieves the similar examples in the corpus when translating.

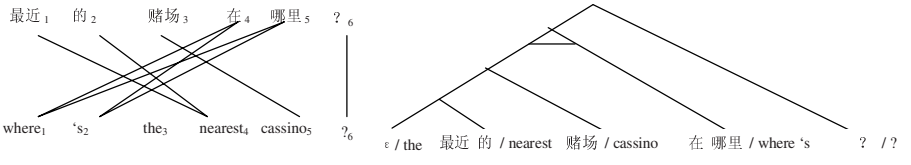
For both of them use the same corpus, we can generate a hybrid MT, which is a SMT system, while using an example-based decoder. Generally, the hybrid MT system uses a word-aligned bilingual corpus, which is a collection of the word-aligned sentence pair, represented as (C, A, E), where C is the source language sentence, and E is the target language sentence and A is the word alignment between C and E.

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 578–585, 2008.
© Springer-Verlag Berlin Heidelberg 2008

A word alignment is defined as follows: given a sentence pair (*C*, *E*), we define a link $l = (i..j, s..t)$, where $i..j$ represents a sequence of words in sentence *C*, where i and j are position indexes, i.e. $i, i+1, ..., j$; and $s..t$ represents a sequence of words in sentence *E*. So each link represents the alignment between the consecutive words in both of the sentences. A word alignment between a sentence pair is a set of links $A = \{l_1, l_2, ..., l_n\}$. In the paper, the example refers to a triple (*C*, *A*, *E*).

We can merge two links $l_1 = (i_1..j_1, s_1..t_1)$ and $l_2 = (i_2..j_2, s_2..t_2)$ to form a larger link, if the two links are adjacent in both of the sentences, i.e. $i_1..j_1$ is adjacent to $i_2..j_2$ where $i_2 = j_1 + 1$ or $i_1 = j_2 + 1$, or $i_1..j_1$ (or $i_2..j_2$) is ε, so do the $s_1..t_1$ to $s_2..t_2$. If the region $(i..j, s..t)$ can be formed by merging two adjacent links gradually, we call the region is **independent**.

Figure 1(a) shows a word alignment example; the number below the word is the position index. In the example, the region (1..3, 3..5) is independent.



(a) An example with word alignment (b) An ITG tree

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

In our hybrid MT system, the word alignment in the corpus must satisfy the Inversion Transduction Grammar (ITG) constraint[3], so that each word alignment will form a binary branching tree, which provides a weak but effective way to constrain the re-ordering of the words. Figure 1b illustrates an ITG tree formed from the alignment.

Using the word-aligned bilingual corpus, we can extract the phrase pairs, called as **blocks**. Each block is formed by combining one or more links, and must be independent. Then, we build the corresponding translation models using the relative frequencies.

3 The Fast Retrieval of Similar Sentences

We divide the retrieval of similar examples into two phases:

- Fast Retrieval Phase: retrieving the similar examples from the corpus quickly, and take them as candidates. The complexity should be not too high.
- Refining Phase: refining the candidates to find the most similar examples.

In this section, we will describe how to retrieve the similar examples quickly through the following three similarity metrics.

3.1 The Matched Words Metric

Given an input sentence $I = w_1 w_2 ... w_n$ and an example (*C*, *A*, *E*), we calculate the number of the matched source words between the input sentence and the source sentence C in the example firstly.

$$Sim_{word}(Input, Example) = \frac{2 * Match_{word}}{Len(I) + Len(C, A, E)} \quad (1)$$

where $Match_{word}$ is the number of the matched words, $Len(I)$ is the number of words in I, and $Len(C, A, E)$ is the number of the words in the in C.

3.2 The Matched Blocks Metric

Given an input sentence $I = w_1 w_2 ... w_n$, we search the blocks for each word $w_i (i \in \{1,2,...n\})$. We use B^i_{k-gram} to represent the blocks, in which for each block (c, e), the source phrase c use the word w_i as the first word, and the length of c is k, i.e. the $c = w_{i..(i+k-1)}$. In order to retrieve the B^i_{k-gram}, we first get all the source phrases with w_i as the first word, and then find the blocks for each source phrase c in the translation model. Considering the complexity, we set *M* as the maximum length of the source phrase (here *M*=3 or 5), i.e., $k \le M$.

For each c, there may exists more than one blocks with c as the source phrase, so we will sort them by the probability and keep the best N (here set N=5) blocks. Now we represent the input sentence as:

$$\sigma(I) = \{block \mid block \in B^i_{k-gram}, 1 \le i \le n, 1 \le k \le M\} \quad (2)$$

For example, in an input sentence "我 来自 中国", the $B^1_{1-gram} = \{(我, i), (我, me), (我, my), (我, Mine)\}$. Note, some B^i_{k-gram} may be empty, e.g. $B^2_{2-gram} = \phi$, since no blocks with "来自 中国" as the source phrase.

In the same way, we represent the example (C, A, E) as:

$$\varphi(C, A, E) = \{block \mid block \in B^i_{k-gram}, block \in A^*, 1 \le k \le M\} \quad (3)$$

where A^* represents the blocks which are links in the alignment A or can be formed by merging adjacent links independently. In order to accelerate the retrieval of similar examples, we generate the block set for each example during the training process and store them in the corpus.

Now, we can use the number of the matched blocks to measure the similarity of the input and the example:

$$Sim_{block}(Input, Example) = \frac{2 * Match_{block}}{B_{gram}^{Input} + B_{gram}^{Example}} \quad (4)$$

where $Match_{block}$ is the number of the matched blocks and B_{gram}^{Input} is the number of B_{k-gram}^{i} ($B_{k-gram}^{i} \neq \phi$) in $\sigma(I)$, and $B_{gram}^{Example}$ is the number of the blocks in $\varphi(C, A, E)$.

Since each block is attached a probability, we can compute the similarity in the following way:

$$Sim_{prob}(Input, Example) = \frac{2 * \sum_{b} \Pr ob(b)}{B_{gram}^{Input} + B_{gram}^{Example}} \quad (5)$$

So the final similarity metric for fast retrieval of the candidates is:

$$Sim_{fast}(Input, Example) = \alpha Sim_{block} + \beta Sim_{word} + \gamma Sim_{prob} \quad (6)$$

where $\alpha + \beta + \gamma = 1$. Zhao et al.[4] provides a method to tune the weights, but here we use mean values, i.e. $\alpha = \beta = \gamma = 1/3$. During the fast retrieval phase, we first filter out the examples using the Sim_{word}, then calculate the Sim_{fast} for each example left, and retrieve the best *N* examples.

4 Refine the Candidates

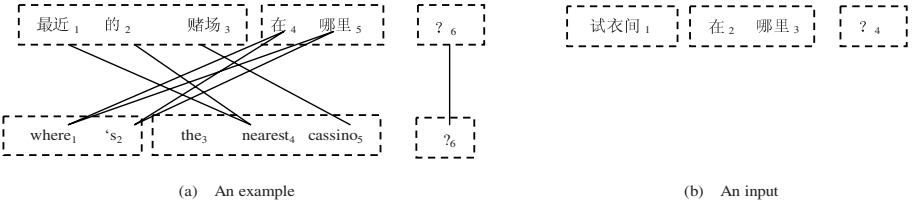
After retrieving the candidate similar examples, we refine the candidates using the swallow structure of the sentences, to find the best *M* similar examples.

4.1 The Alignment Structure Metric

Given the input sentence I and an example (C, A, E), we first search the matched blocks, at this moment the order of the source phrases in the blocks must correspond with the order of the words in the input.

As Figure 2 shows, the matching divides the input and the example respectively into several regions, where some regions are matched and some un-matched. And we take each region as a whole and align them between the input and the example according to the order of the matched regions. For example, the region (1..3,3..5) in (C, A, E) is un-matched, which aligns to the region (1..1) in I. In this way, we can use a similar edit distance method to measure the similarity. We count the number of the Deletion/Insertion/Substitution operations, which take the region as the object.

We set the penalty for each deletion and insertion operation as 1, while considering the un-matched region in the example may be independent or not, we set the penalty for substitution as 0.5 if the region is independent, otherwise as 1. E.g., the distance is 0.5 for substituting the region (1..3,3..5) to (1..1).



(a) An example (b) An input

Fig. 2. An input and an example. After matching, there are three regions in both sides, which are included in the line box, where the region (4..5,1..2) in the example matches the region (2..3) in the input, so do (6..6,6..6) to (4..4). And the region (1..3,3..5) in the example should be substituted to (1..1) in the input.

We get the metric for measuring the structure similarity of the I and (C, A, E):

$$Sim_{align}(Input, Example) = 1 - \frac{D + I + S}{R_{input} + R_{exmaple}} \tag{7}$$

where D, I, S are the deletion, insertion and substitution distances, respectively. And the R_{input} and $R_{exmaple}$ are the region numbers in the input and example.

4.2 The Semantic Metric

We calculate the semantic distance using the "Tong_Yi_Ci_Ci_Lin"[6], which is a Chinese semantic lexicon. In the lexicon, the semantics of the words are divided into three levels, called large, middle and small level, represented as the uppercase letter, lowercase letter, and integer number respectively, so that it can represent a semantic using a unique code. Each code may include one or more words, and each word may have one or more codes, for it may have multiple semantics. For example, "我" and "咱" have the same code "Aa02", the code for "你" is "Aa03".

Now, we can compute the semantic distance between two words using the codes they are marked: If the large level is different, the distance is 3; the middle is 2; and the small is 1; if the codes are equal, the distance is 0. Since each word has more than one code, we will choose the nearest distance.

Using the semantic distance of words, we can measure the semantic distance of two regions in the input and the example.

$$Dist_{reg\_sem}(reg_1, reg_2) = \frac{\sum_{w_1 \in reg_1} \min_{w_2 \in reg_2} [Dist_{word\_sem}(w_1, w_2)]/3}{Len(reg_1) + Len(reg_2)} \tag{8}$$

Given the same match as section 4.1, the substitution distance is the sum of semantic distance of the substitution regions:

$$S_{sem}(Input, Example) = \sum_{regions} Dist_{reg\_sem}(region_{input}, region_{example}) \tag{9}$$

Then, we obtain the semantic metric:

$$Sim_{sem}(Input, Example) = 1 - \frac{D + I + S_{sem}}{R_{input} + R_{exmaple}} \tag{10}$$

In the end, we obtain the similarity metric, which considers all of the above metrics:

$$Sim_{final}(Input, Example) = \alpha' Sim_{fast} + \beta' Sim_{align} + \gamma' Sim_{sem} \tag{11}$$

where $\alpha'+\beta'+\gamma'=1$. Here we also use the mean values as the weights.

5 Experiments

We carried out experiments on a Chinese-English bilingual corpus, which is the training corpus in the open Chinese-English translation task of IWSLT2007, consisting of the sentence-aligned spoken language text for traveling. The training corpus consists of 39,953 sentence pairs, and the test set 489 Chinese sentences. We lowercased and stemmed the words in the English sentences for preprocessing.

During the training, we obtained firstly the word alignments, which satisfy the ITG constraint, for the sentence pairs in the corpus; then built the translation models, and retrieved and stored the blocks in each examples, i.e. $\varphi(C, A, E)$.

5.1 Evaluation of the Retrieval of Similar Examples

We evaluate our retrieval method in manual way: obtain 200 input sentences from the test set randomly, take them as inputs to retrieve the similar examples (10 examples for each input), and verify manually whether the retrieved examples are similar to the inputs.

We define a correct similar example as the following way: if the retrieved example can be turned to the input sentence by modifying x independent regions, where each region consists of at most y blocks (here set $y = 2$).

To evaluate the two retrieval phases, we also get the best 10 examples for the fast retrieval phrase in the candidates, and compare with the final 10 examples. Table 1 shows the results. It shows the correct number and accuracies for both phases by changed the x. With x becomes larger, the accuracies are higher. When $x = 1$, the accuracies are both 45%, because some of the input sentences are very short, and both of the two phases will find the exactly similar examples. But when x becomes larger, the refined retrieval will get more accurate examples.

5.2 Evaluation of the Translation

Our goal is to improve the translation quality, so we combined the retrieval module into our hybrid MT system, and compared the translation results with other SMT systems. The results list in Table 2.

The first column lists the MT systems: the first two systems are two SMT systems, where Moses[7] is a state-of-the-art SMT system, and SMT-CKY is our SMT system

with a CKY style decoder; the last two systems are hybrid MT systems with fast retrieval module and refined retrieval module. The second column lists the Bleu scores[8] for these systems. The results show that our hybrid MT system achieves an improvement when using the new retrieval module. We conclude our method to retrieve the similar examples is effective.

Table 1. The results for retrieval of similar sentences

	Test Number	$x=1$		$x=2$		$x=3$	
		Corr.	Accu.	Corr.	Accu.	Corr.	Accu.
Fast Retrieval	200	90	45%	141	70.5%	155%	77.5%
Refined Retrieval	200	90	45%	153	76.5%	174	87%

Table 2. The results of translation

System	Bleu(%)
Moses	22.61
SMT-CKY	28.33
Hybrid MT with fast retrieval	30.03
Hybrid MT with refined retrieval	33.05

6 Conclusions

We present a method to retrieve the similar examples for the input sentence in the hybrid MT systems. The approach makes good use of the word alignment information contained in the corpus. Since the method considers the internal translation process contained implicitly in each example, it helps to improve the translation quality.

In the future, we will improve the refining metric further, especially the semantic metric, and consider how to tune the weights. Also, we should consider how to use the retrieved examples more effectively.

Acknowledgements

This work is supported by the National Science Foundation of China under Grants No. 60573057, 60473057 and 90604007.

References

1. Mandreoli, F., Martoglia, R., Tiberio, P.: Searching Similar (Sub)Sentences for Example-Based Machine Translation. In: Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Datt (SEBD 2002), Isola d'Elba, Italy (2002)
2. Watanabe, T., Sumita, E.: Example-based Decoding for Statistical Machine Translation. Machine Translation Summit IX, 410–417 (2003)

3. Wu, D.: Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics 23(3), 374 (1997)
4. Zhao, Y.Y., Qin, B., Liu, T., Su, Z.-Z.: Sentence Similarity Computing Based on Multi-Features Fusion. In: The Proceedings of JSCL 2005, August 2005, pp. 168–174 (2005)
5. Li, B., Liu, T., Qin, B., Li, S.: Chinese Sentence Similarity Computing Based on Semantic Dependency Parsing (2003)
6. Mei, J.J., Zhu, Y.M., Gao, Y.Q., Yin, H.X.: Tong Yi Ci Ci Lin, 2nd edn., Shanghai (1996)
7. Moses, http://www.statmt.org/moses/
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311–318 (2002)