Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

A MODEL FOR RANKING SENTENCE PAIRS IN PARALLEL CORPORA

YI-DONG CHEN, XIAO-DONG SHI, CHANG-LE ZHOU, QING-YANG HONG

Department of Computer Science, Xiamen University, Xiamen, Fujian, P. R. China, 361005
E-MAIL: {ydchen, mandel, dozero, qyhong}@xmu.edu.cn

Abstract:

In this paper, the problem of ranking sentence pairs in parallel corpora was addressed for the first time. To solve this problem, a novel model was proposed. In this model, both syntax features and semantics features of sentence pairs are considered. Since most today's Statistical Machine Translation models depend on word alignment, features related to word alignment information are also included. Two experiments were carried out and the results showed that the model had promising performance.

Keywords:

Ranking; parallel corpora; corpus revision; statistical machine translation

1. Introduction

1.1. Motivation

In the past ten years, statistical methods have been more and more popular in the research of Machine Translation (MT). The performance of a Statistical Machine Translation (SMT) system is dependent on many aspects, such as the translation model, the search strategy and the parallel corpus. Specifically, parallel corpus has become an essential resource for the SMT system.

It's time-consuming to construct parallel corpus manually. Thus, more and more researchers choose to extract parallel text from Web or comparable corpus through automatic way, such as the study of [1] and [2]. Since there are many errors in the Web and the program used to extract parallel text are little-intelligent, the automatic way often results in large-scale but low-quality corpora. Therefore, an automatically constructed corpus always requires careful revision before it could be used in a SMT system. Unfortunately, the corpus revising tasks are tiring and time-consuming. So it will be useful if some mechanics can be developed to help performing the revision tasks. This paper addresses this problem and proposes a novel model.

1.2. Basic Idea

There are several ways to help revising a corpus. Our method follows a simple idea, that is, to design a ranking model which will rank the sentence pairs in the corpus so that the *better* sentence pairs are closer to the front. In principle, a sentence pair will be considered as *good* if it satisfies all the following three conditions:

Con. 1: there are little errors in the sentence pair.

Con. 2: the sentence pair is a good translation pair.

Con. 3: the sentence pair will make contributions for later usages in SMT training models.

As an example, the following sentence pair shows how a sentence pair satisfies the first two conditions well while violates the last condition.

Source (Chinese, in pinyin): *Wan Nai Ju Ji*

Target: *everywhere seemed silent*

Based on such an idea, most sentence pairs containing idioms will be looked on as not so good. This is quite natural, since the learning model of today's SMT systems are too simple to learn useful knowledge from such sentence pairs.

The rest of this paper is organized as follows. In Section 2, we bring forward the framework of our ranking model. In Section 3, we describe in detail the feature incorporated in the ranking model. In Section 4, we discuss about the parameters. In Section 5, two experiments are conducted and the results are reported.

2. Ranking Framework

The problem of ranking a parallel corpus could be solved through the following two steps:

Step 1: compute a numerical metric for each sentence pair in the corpus.

Step 2: rank the sentence pairs according to the metrics assigned to them.

Following the idea in Section 1.2, the number metric should be able to embody how *good* a sentence pair is, which could be examined from many different aspects. We

0-7803-9091-1/05/$20.00 ©2005 IEEE

call each aspect a feature, and calculate the metric as follows.

$$RM(f,e)=\sum_{i=1}^{n}\lambda_i \cdot F_i(f,e) \quad (1)$$

Here, *f* and *e*, represent source sentence and target sentence, respectively. The function $F_i(f, e)$ gives the score of the *i*th feature, and the parameter λ_i is the corresponding weight.

3. Features

3.1. Language Model Scores

The use of language models in SMT could be traced back to [3]. SMT models use the language model factors to indicate how *natural* and *grammatical* the output sentence is. In our ranking model, we incorporate language model score as a feature, named LM, for the same purpose.

$$LM(f,e)=\frac{\sqrt[len(f)]{p(f)}+\sqrt[len(e)]{p(e)}}{2} \quad (2)$$

In Formula 2, the language model factor is calculated for each sentence respectively, and the function *len*(*s*) that will return the length of a given sentence *s* is used to eliminate the influence of the sentence length.

3.2. Length Consistency Scores

The sentence length consistency is a basic factor considered in sentence aligning model such as [4]. So for a corpus which is constructed automatically, sentences in each sentence pairs should be consistent with each other in length. However, according to our observation, there still some sentence pairs violate the length consistency. In other words, sentences in such sentence pairs have quite different length. Most of these sentence pairs contain errors. To assign such sentence pairs low mark, we introduce a feature called LC. Formula 3 shows how to compute this feature.

$$LC(f,e)=\begin{cases}1 & \alpha_1 \cdot len(e) \le len(f) \le \alpha_2 \cdot len(e) \\ 0 & otherwise\end{cases} \quad (3)$$

Here, the function *len*(*s*) is the same as the one in Section 3.1. Parameter α_1 and α_2 could be modified according to the language pair. In our system, they are set as 0.5 and 1.2, respectively. This system is used to rank a Chinese-English corpus.

3.3. bPER Scores

This feature is introduced to determinate how likely a given sentence pair is a translation pair. The main idea is that *the more words of a given sentence could find their translations in the other sentence, the more likely the given sentence is the translation of the other one.* Based on such an idea, most sentence pairs with translation errors or containing idioms will get low marks. This feature is similar, conceptually, to the position-independent word error rate (PER) metric proposed in [5], except that we evaluate one sentence using the sentence in other language. For this reason, we call it bPER, where b- stands for bilingual. In the following, Formula 4, 5 and 6 describe how this feature could be computed.

$$bPER(f,e)=\frac{|T(f,e)|}{|I(f)|}\times\frac{|T(e,f)|}{|I(e)|} \quad (4)$$

$$I(s)=\{w \mid w\in s \wedge pos(w)\in\{n,v,adj,perp\}\} \quad (5)$$

$$T(s_1,s_2)=\{w \mid w\in I(s_1)\wedge \exists v(v\in s_2 \wedge t(w,v))\} \quad (6)$$

Here, the function *pos*(*w*) returns the Part-of-Speech tag of a given word *w*, and the function *t*(*w*, *v*) checks whether or not two given words, *w* and *v*, are translations of each other.

Note that when calculating this feature, we consider only some kinds of so called *important* words in the given sentences rather than all the words. The reason is that most of the rest kinds of words are functional words, which usually have no corresponding words when translating.

3.4. bWER Scores

This feature is introduced for the same purpose as bPER scores feature in Section 3.3, and it is very similar to the previous one except that it takes the word alignment information into account. We decided to incorporate this feature with the thought that it could be useful since current Statistical Machine Translation models all depends on word alignment.

We call this feature bWER just to show that it is more similar to word error rate (WER) metric of [7]. In Formula 7, 8 and 9, we demonstrate how to calculate this feature in detail. Note that before such calculation could be conducted, the given sentence pair should be aligned using some word alignment method such as the IBM Models presented in [3].

$$bWER(f,e)=\frac{|TA(f,e)|}{|IA(f,e)|} \quad (7)$$

$$IA(s_1,s_2)=\{(w,v) \mid w\in I(s_1)\wedge v\in I(s_2)\wedge w \ and \ v \ are \ aligned\} \quad (8)$$

$$T(s_1,s_2)=\{(w,v) \mid (w,v)\in IA(s_1,s_2)\wedge t(w,v)\} \quad (9)$$

Here, the function $I(s)$ and $t(w, v)$ are the same as the description in Section 3.3.

In theory, this feature is more accurate than the one in 3.3 because it concerned the alignment information. But in practice, the word alignment model used is not so accurate. So in our ranking model, this feature was assigned a lower weight.

4. Parameters

Given the framework in Section 2 and the features in Section 3, we have a ranking model with the weight parameters undetermined. These parameters could be trained more accurately from a training set or through a bootstrapping way. For simplicity, we just set them empirically, as shown in Table 1.

Table 1. Parameter Values

Parameters	Corresponding Features	Values
λ_1	Language Model Score	0.2
λ_2	Length Consistency Score	0.1
λ_3	bPER Score	0.5
λ_4	bWER Score	0.2

5. Experiments

It's difficult to evaluate our ranking model since there is no test set that contains ranked parallel texts existed before, and to construct such a test set is not very easy. In this section, two experiments that evaluate our model in tricky way are reported. And both the results show that our ranking model has good performance.

The first experiment is carried out to test how likely the result of our ranking model is consistent with the human's result without requiring the person to rank the sentence pairs.

The second experiment demonstrates one of the usages of our ranking model. The performance of the ranking model could then be learned, to some extent, from the result indirectly.

5.1. Consistency with the Human's Viewpoint

Firstly, a test set with 300 sentence pairs selected randomly from a parallel corpus was constructed.

Then three persons were required to evaluate the sentence pairs in the test set and assigned each sentence pair a 2-valued, *good* or *bad*, tag. After this step, we got three evaluated reference set, called R_1, R_2 and R_3 respectively.

And then, our ranking model was performed upon the test set and output a ranked set, called R.

Finally, we evaluated R using Formula 10 and 11.

$$ErrorRate = \frac{|ESet|}{C_{|R|}^2} \tag{10}$$

$$\begin{aligned} ESet = \{(sp_1, sp_2) \mid & sp_1, sp_2 \in R \wedge \\ & sp_1 \neq sp_2 \wedge \\ & pre(sp_1, sp_2) \wedge \\ & \bigvee_{i=1}^{3} (bad(sp_1, R_i) \wedge good(sp_2, R_i))\} \end{aligned} \tag{11}$$

Here, the function $pre(sp_1, sp_2)$ checks whether or not the given sentence pair sp_1 is before the second one sp_2 in the ranked set R, and the function $bad(sp, R_i)$ and $good(sp, R_i)$ checks respectively whether the given sentence pair sp is labeled *bad* or *good* in the given reference set R_i.

It's clear that, the smaller the *ErrorRate* is, the better the performance of our ranking model is. And we finally got the *ErrorRate* of 0.04, which show that the ranking model could be performed well.

5.2. One of the Usages

Firstly, we performed our ranking model upon a parallel corpus with 80 thousand sentence pairs and achieved a ranked parallel corpus.

Secondly, we constructed five subsets, all containing 40 thousand sentence pairs of the ranked corpus. Some comments on how these subsets were constructed are listed in Table 2.

Table 2. Five Subsets of the Ranked Corpus

Names	Comments
G	The sentence pairs were extracted from the first part of the ranked corpus.
B	The sentence pairs were extracted from the last part of the ranked corpus.
R_1	The sentence pairs were extracted from the ranked corpus randomly.
R_2	The sentence pairs were extracted from the ranked corpus randomly, too.
R_3	Again, the sentence pairs were extracted from the ranked corpus randomly.

Thirdly, we used the subsets as training sets, respectively, to train our SMT engine and got five different SMT systems.

Finally, we ran the five systems respectively upon the

test data from the *2004 China's National 863 MT Evaluation*. We then evaluated their results using both BLEU metric and NIST metric.

The final evaluation results are listed in Table 3.

Table 3. MT Evaluation Results

	G	B	R_1	R_2	R_3
BLEU	0.0574	0.0527	0.0553	0.0554	0.0540
NIST	3.9851	3.6435	3.8755	3.8832	3.8545

Note that due to the low quantity of the training set, the scores of the MT systems are really low as shown in Table 2. But the results clearly show that the system using training set *G* outperformed all the other systems. And this means that the sentence pairs in *G* are better as a whole.

6. Conclusions

In this paper, we have proposed a model to rank sentence pairs in parallel corpus. Based on such a ranking model, one may carry out the corpus revision task more easily. Two experiments have been performed, and both the results show that our ranking model has promising performance. The result of the first experiment shows that the ranking output of our ranking model is consistent with the human's viewpoints, and thus indicates that the model is reasonable. In the second experiment, we successfully demonstrate one potential usage of our ranking model, i.e., ranking a given training corpus and stripping out the last part and then using the result corpus directly to train a SMT engine.

Acknowledgements

This work was supported by the Chinese 863 High Tech Research Fund (2004AA117010), and the Chinese National Natural Science Fund (60373080).

References

[1] Philip Resnik, and Noah A. Smith, "The Web as a Parallel Corpus", Computational Linguistics, Vol 29, No. 3, pp. 349-380, Sep. 2003.

[2] Dragos S. Munteanu, Alexander Fraser, and Daniel Marcu, "Improved machine translation performance via parallel sentence extraction from comparable corpora", Proceeding of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004), Boston, MA, pp. 265-272, May 2004.

[3] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robet L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", Computational Linguistics, Vol 19, No. 2, pp. 263-311, 1993.

[4] William A. Gale, and Kenneth W. Church, "A Program for Aligning Sentences in Bilingual Corpora", Computational Linguistics, Vol 19, No. 1, pp. 75-102, Mar. 1993.

[5] Christoph Tillmann, Stephan Vogel, Hermann Ney, Alex Zubiaga, and Hassan Sawaf, "Accelerated DP based Search for Statistical Translation", Proceeding of European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 2667-2670, Sep. 1997.

[6] Sonja Nieβen, Franz J. Och, G. Leusch, and Hermann Ney, "An Evaluation Tool for Machine Translation: Fast Evaluation for Machine Translation Research", Proceeding of the Second International Conference on Language Resources and Evaluation (LREC), Athens, Greece, pp. 39-45, May 2000.