

A New Approach for English-Chinese Named Entity Alignment

Donghui Feng*

Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina Del Rey, CA, U.S.A, 90292
donghui@isi.edu

Yajuan Lv†

†Microsoft Research Asia
5F Sigma Center, No.49 Zhichun Road, Haidian
Beijing, China, 100080
{t-yjlv, mingzhou}@microsoft.com

Ming Zhou†

Abstract

Traditional word alignment approaches cannot come up with satisfactory results for Named Entities. In this paper, we propose a novel approach using a maximum entropy model for named entity alignment. To ease the training of the maximum entropy model, bootstrapping is used to help supervised learning. Unlike previous work reported in the literature, our work conducts bilingual Named Entity alignment without word segmentation for Chinese and its performance is much better than that with word segmentation. When compared with IBM and HMM alignment models, experimental results show that our approach outperforms IBM Model 4 and HMM significantly.

1 Introduction

This paper addresses the Named Entity (NE) alignment of a bilingual corpus, which means building an alignment between each source NE and its translation NE in the target language. Research has shown that Named Entities (NE) carry essential information in human language (Hobbs et al., 1996). Aligning bilingual Named Entities is an effective way to extract an NE translation list and translation templates. For example, in the following sentence pair, aligning the NEs, [Zhi Chun road] and [知春路] can produce a translation template correctly.

- Can I get to [LN Zhi Chun road] by eight o'clock?
- 八点我能到 [LN 知春路]吗?

In addition, NE alignment can be very useful for Statistical Machine Translation (SMT) and Cross-Language Information Retrieval (CLIR).

A Named Entity alignment, however, is not easy to obtain. It requires both Named Entity Recognition (NER) and alignment be handled correctly. NEs may not be well recognized, or only

parts of them may be recognized during NER. When aligning bilingual NEs in different languages, we need to handle many-to-many alignments. And the inconsistency of NE translation and NER in different languages is also a big problem. Specifically, in Chinese NE processing, since Chinese is not a tokenized language, previous work (Huang et al., 2003) normally conducts word segmentation and identifies Named Entities in turn. This involves several problems for Chinese NEs, such as word segmentation error, the identification of Chinese NE boundaries, and the mis-tagging of Chinese NEs. For example, “国防部长” in Chinese is really one unit and should not be segmented as [ON 国防部]/长. The errors from word segmentation and NER will propagate into NE alignment.

In this paper, we propose a novel approach using a maximum entropy model to carry out English-Chinese Named Entity¹ alignment. NEs in English are first recognized by NER tools. We then investigate NE translation features to identify NEs in Chinese and determine the most probable alignment. To ease the training of the maximum entropy model, bootstrapping is used to help supervised learning.

On the other hand, to avoid error propagations from word segmentation and NER, we directly extract Chinese NEs and make the alignment from plain text without word segmentation. It is unlike previous work reported in the literature. Although this makes the task more difficult, it greatly reduces the chance of errors introduced by previous steps and therefore produces much better performance on our task.

To justify our approach, we adopt traditional alignment approaches, in particular IBM Model 4 (Brown et al., 1993) and HMM (Vogel et al., 1996), to carry out NE alignment as our baseline systems. Experimental results show that in this task our approach outperforms IBM Model 4 and HMM significantly. Furthermore, the performance

* The work was done while the first author was visiting Microsoft Research Asia.

¹ We only discuss NEs of three categories: Person Name (PN), Location Name (LN), and Organization Name (ON).

without word segmentation is much better than that with word segmentation.

The rest of this paper is organized as follows: In section 2, we discuss related work on NE alignment. Section 3 gives the overall framework of NE alignment with our maximum entropy model. Feature functions and bootstrapping procedures are also explained in this section. We show experimental results and compare them with baseline systems in Section 4. Section 5 concludes the paper and discusses ongoing future work.

2 Related Work

Translation knowledge can be acquired via word and phrase alignment. So far a lot of research has been conducted in the field of machine translation and knowledge acquisition, including both statistical approaches (Cherry and Lin, 2003; Probst and Brown, 2002; Wang et al., 2002; Och and Ney, 2000; Melamed, 2000; Vogel et al., 1996) and symbolic approaches (Huang and Choi, 2000; Ker and Chang, 1997).

However, these approaches do not work well on the task of NE alignment. Traditional approaches following IBM Models (Brown et al., 1993) are not able to produce satisfactory results due to their inherent inability to handle many-to-many alignments. They only carry out the alignment between words and do not consider the case of complex phrases like some multi-word NEs. On the other hand, IBM Models allow at most one word in the source language to correspond to a word in the target language (Koehn et al., 2003; Marcu, 2001). Therefore they can not handle many-to-many word alignments within NEs well. Another well-known word alignment approach, HMM (Vogel et al., 1996), makes the alignment probabilities depend on the alignment position of the previous word. It does not explicitly consider many-to-many alignment either.

Huang et al. (2003) proposed to extract Named Entity translanguagual equivalences based on the minimization of a linearly combined multi-feature cost. But they require Named Entity Recognition on both the source side and the target side. Moore's (2003) approach is based on a sequence of cost models. However, this approach greatly relies on linguistic information, such as a string repeated on both sides, and clues from capital letters that are not suitable for language pairs not belonging to the same family. Also, there are already complete lexical compounds identified on the target side, which represent a big part of the final results. During the alignment, Moore does not hypothesize that translations of phrases would require splitting predetermined lexical compounds on the target set.

These methods are not suitable for our task, since we only have NEs identified on the source side, and there is no extra knowledge from the target side. Considering the inherent characteristics of NE translation, we can find several features that can help NE alignment; therefore, we use a maximum entropy model to integrate these features and carry out NE alignment.

3 NE Alignment with a Maximum Entropy Model

Without relying on syntactic knowledge from either the English side or the Chinese side, we find there are several valuable features that can be used for Named Entity alignment. Considering the advantages of the maximum entropy model (Berger et al., 1996) to integrate different kinds of features, we use this framework to handle our problem.

Suppose the source English NE ne_e , $ne_e = \{e_1, e_2 \dots e_n\}$, consists of n English words and the candidate Chinese NE ne_c , $ne_c = \{c_1, c_2 \dots c_m\}$, is composed of m Chinese characters. Suppose also that we have M feature functions $h_m(ne_c, ne_e)$, $m = 1, \dots, M$. For each feature function, we have a model parameter λ_m , $m = 1, \dots, M$. The alignment probability can be defined as follows (Och and Ney, 2002):

$$P(ne_c | ne_e) = p_{\lambda^M}(ne_c | ne_e) \\ = \frac{\exp[\sum_{m=1}^M \lambda_m h_m(ne_c, ne_e)]}{\sum_{ne'_c} \exp[\sum_{m=1}^M \lambda_m h_m(ne'_c, ne_e)]} \quad (3.1)$$

The decision rule to choose the most probable aligned target NE of the English NE is (Och and Ney, 2002):

$$\hat{ne}_c = \arg \max_{ne_c} \{P(ne_c | ne_e)\} \\ = \arg \max_{ne_c} \left\{ \sum_{m=1}^M \lambda_m h_m(ne_c, ne_e) \right\} \quad (3.2)$$

In our approach, considering the characteristics of NE translation, we adopt 4 features: translation score, transliteration score, the source NE and target NE's co-occurrence score, and distortion score for distinguishing identical NEs in the same sentence. Next, we discuss these four features in detail.

3.1 Feature Functions

3.1.1 Translation Score

It is important to consider the translation probability between words in English NE and characters in Chinese NE. When processing

Chinese sentence without segmentation, word here refers to single Chinese character.

The translation score here is used to represent how close an NE pair is based on translation probabilities. Supposing the source English NE ne_e consists of n English words, $ne_e = \{e_1, e_2 \dots e_n\}$ and the candidate Chinese NE ne_c is composed of m Chinese characters, $ne_c = \{c_1, c_2 \dots c_m\}$, we can get the translation score of these two bilingual NEs based on the translation probability between e_i and c_j :

$$S(ne_e, ne_c) = \sum_{j=1}^m \sum_{i=1}^n p(c_j | e_i) \quad (3.3)$$

Given a parallel corpus aligned at the sentence level, we can achieve the translation probability between each English word and each Chinese character $p(c_j | e_i)$ via word alignments with IBM Model 1 (Brown et al., 1993). Without word segmentation, we have to calculate every possible candidate to determine the most probable alignment, which will make the search space very large. Therefore, we conduct pruning upon the whole search space. If there is a score jump between two adjacent characters, the candidate will be discarded. The scores between the candidate Chinese NEs and the source English NE are calculated via this formula as the value of this feature.

3.1.2 Transliteration Score

Although in theory, translation scores can build up relations within correct NE alignments, in practice this is not always the case, due to the characteristics of the corpus. This is more obvious when we have sparse data. For example, most of the person names in Named Entities are sparsely distributed in the corpus and not repeated regularly. Besides that, some English NEs are translated via transliteration (Lee and Chang, 2003; Al-Onaizan and Knight, 2002; Knight and Graehl, 1997) instead of semantic translation. Therefore, it is fairly important to make transliteration models.

Given an English Named Entity e , $e = \{e_1, e_2 \dots e_n\}$, the procedure of transliterating e into a Chinese Named Entity c , $c = \{c_1, c_2 \dots c_m\}$, can be described with Formula (3.4) (For simplicity of denotation, we here use e and c to represent English NE and Chinese NE instead of ne_e and ne_c).

$$\hat{c} = \arg \max_c P(c | e) \quad (3.4)$$

According to Bayes' Rule, it can be transformed to:

$$\hat{c} = \arg \max_c P(c) * P(e | c) \quad (3.5)$$

Since there are more than 6k common-used Chinese characters, we need a very large training corpus to build the mapping directly between English words and Chinese characters. We adopt a romanization system, Chinese PinYin, to ease the transformation. Each Chinese character corresponds to a Chinese PinYin string. And the probability from a Chinese character to PinYin string is $P(r | c) \approx 1$, except for polyphonous characters. Thus we have:

$$\hat{c} = \arg \max_c P(c) * P(r | c) * P(e | r) \quad (3.6)$$

Our problem is: Given both English NE and candidate Chinese NEs, finding the most probable alignment, instead of finding the most probable Chinese translation of the English NE. Therefore unlike previous work (Lee and Chang, 2003; Huang et al., 2003) in English-Chinese transliteration models, we transform each candidate Chinese NE to Chinese PinYin strings and directly train a PinYin-based language model with a separate English-Chinese name list consisting of 1258 name pairs to decode the most probable PinYin string from English NE.

To find the most probable PinYin string from English NE, we rewrite Formula (3.5) as the following:

$$\hat{r} = \arg \max_r P(r) * P(e | r) \quad (3.7)$$

where r represents the romanization (PinYin string), $r = \{r_1, r_2 \dots r_m\}$. For each of the factor, we have

$$P(e | r) = \prod_{i=1}^m P(e_i | r_i) \quad (3.8)$$

$$P(r) = P(r_1)P(r_2 | r_1) \prod_{i=3}^m P(r_i | r_{i-2}r_{i-1}) \quad (3.9)$$

where e_i is an English syllable and r_i is a Chinese PinYin substring.

For example, we have English NE "Richard" and its candidate Chinese NE "理查德". Since both the channel model and language model are PinYin based, the result of Viterbi decoding is from "Richard" to "Li Cha De". We transform "理查德" to the PinYin string "Li Cha De". Then we compare the similarity based on the PinYin string instead of with Chinese characters directly. This is because when transliterating English NEs into Chinese, it is very flexible to choose which character to simulate the pronunciation, but the PinYin string is relatively fixed.

For every English word, there exist several ways to partition it into syllables, so here we adopt a dynamic programming algorithm to decode the English word into a Chinese PinYin sequence. Based on the transliteration string of the English

NE and the PinYin string of the original candidate Chinese NE, we can calculate their similarity with the XDice coefficient (Brew and McKelvie, 1996). This is a variant of Dice coefficient which allows “extended bigrams”. An extended bigram (*xbig*) is formed by deleting the middle letter from any three-letter substring of the word in addition to the original bigrams.

Suppose the transliteration string of the English NE and the PinYin string of the candidate Chinese NE are e_{il} and c_{py} , respectively. The XDice coefficient is calculated via the following formula:

$$XDice(e_{il}, c_{py}) = \frac{2 \times |xbigs(e_{il}) \cap xbigs(c_{py})|}{|xbigs(e_{il})| + |xbigs(c_{py})|} \quad (3.10)$$

Another point to note is that foreign person names and Chinese person names have different translation strategies. The transliteration framework above is only applied on foreign names. For Chinese person name translation, the surface English strings are exactly Chinese person names’ PinYin strings. To deal with the two situations, let e_{sur} denote the surface English string, the final transliteration score is defined by taking the maximum value of the two XDice coefficients:

$$Tl(c, e) = \max(XDice(c_{py}, e_{il}), XDice(c_{py}, e_{sur})) \quad (3.11)$$

This formula does not differentiate foreign person names and Chinese person names, and foreign person names’ transliteration strings or Chinese person names’ PinYin strings can be handled appropriately. Besides this, since the English string and the PinYin string share the same character set, our approach can also work as an alternative if the transliteration decoding fails.

For example, for the English name “Cuba”, the alignment to a Chinese NE should be “古巴”. If the transliteration decoding fails, its PinYin string, “Guba”, still has a very strong relation with the surface string “Cuba” via the XDice coefficient. This can make the system more powerful.

3.1.3 Co-occurrence Score

Another approach is to find the co-occurrences of source and target NEs in the whole corpus. If both NEs co-occur very often, there exists a big chance that they align to each other. The knowledge acquired from the whole corpus is an extra and valuable feature for NE alignment. We calculate the co-occurrence score of the source English NE and the candidate Chinese NE with the following formula:

$$P_{co}(ne_c | ne_e) = \frac{count(ne_c, ne_e)}{\sum count(*, ne_e)} \quad (3.12)$$

where $count(ne_c, ne_e)$ is the number of times ne_c and ne_e appear together and $count(*, ne_e)$ is the number of times that ne_e appears. This probability is a good indication for determining bilingual NE alignment.

3.1.4 Distortion Score

When translating NEs across languages, we notice that the difference of their positions is also a good indication for determining their relation, and this is a must when there are identical candidates in the target language. The bigger the difference is, the less probable they can be translations of each other. Therefore, we define the distortion score between the source English NE and the candidate Chinese NE as another feature.

Suppose the index of the start position of the English NE is i , and the length of the English sentence is m . We then have the relative position of

the source English NE $pos_e = \frac{i}{m}$, and the candidate Chinese NE’s relative position pos_c , $0 \leq pos_e, pos_c \leq 1$. The distortion score is defined with the following formula:

$$Dist(ne_c, ne_e) = 1 - ABS(pos_e - pos_c) \quad (3.13)$$

where ABS means the absolute value. If there are multiple identical candidate Chinese NEs at different positions in the target language, the one with the largest distortion score will win.

3.2 Bootstrapping with the MaxEnt Model

To apply the maximum entropy model for NE alignment, we process in two steps: selecting the NE candidates and training the maximum entropy model parameters.

3.2.1 NE Candidate Selection

To get an NE alignment with our maximum entropy model, we first use NLPWIN (Heidorn, 2000) to identify Named Entities in English. For each word in the recognized NE, we find all the possible translation characters in Chinese through the translation table acquired from IBM Model 1. Finally, we have all the selected characters as the “seed” data. With an open-ended window for each seed, all the possible sequences located within the window are considered as possible candidates for NE alignment. Their lengths range from 1 to the empirically determined length of the window. During the candidate selection, the pruning strategy discussed above is applied to reduce the search space.

For example, in Figure 1, if “China” only has a translation probability over the threshold value with “中”, the two seed data are located with the index of 0 and 4. Supposing the length of the window to be 3, all the candidates around the seed data including “中国”, with the length ranging from 1 to 3, are selected.

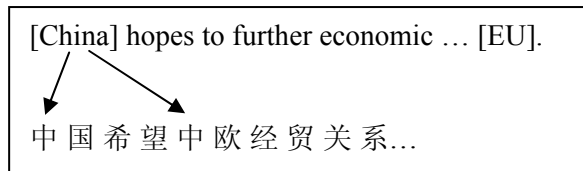


Figure 1. Example of Seed Data

3.2.2 MaxEnt Parameter Training

With the four feature functions defined in Section 3.1, for each identified NE in English, we calculate the feature scores of all the selected Chinese NE candidates.

To achieve the most probable aligned Chinese NE, we use the published package YASMET² to conduct parameter training and re-ranking of all the NE candidates. YASMET requires supervised learning for the training of the maximum entropy model. However, it is not easy to acquire a large annotated training set. Here bootstrapping is used to help the process. Figure 2 gives the whole procedure for parameter training.

1. Set the coefficients λ_i as uniform distribution;
2. Calculate all the feature scores to get the N-best list of the Chinese NE candidates;
3. Candidates with their values over a given threshold are considered to be correct and put into the re-ranking training set;
4. Retrain the parameters λ_i with YASMET;
5. Repeat from Step 2 until λ_i converge, and take the current ranking as the final result.

Figure 2. Parameter Training

4 Experimental Results

4.1 Experimental Setup

We perform experiments to investigate the performance of the above framework. We take the LDC Xinhua News with aligned English-Chinese sentence pairs as our corpus.

The incremental testing strategy is to investigate the system’s performance as more and more data are added into the data set. Initially, we take 300

sentences as the standard testing set, and we repeatedly add 5k more sentences into the data set and process the new data. After iterative re-ranking, the performance of alignment models over the 300 sentence pairs is calculated. The learning curves are drawn from 5k through 30k sentences with the step as 5k every time.

4.2 Baseline System

A translated Chinese NE may appear at a different position from the corresponding English NE in the sentence. IBM Model 4 (Brown et al., 1993) integrates a distortion probability, which is complete enough to account for this tendency. The HMM model (Vogel et al., 1996) conducts word alignment with a strong tendency to preserve localization from one language to another. Therefore we extract NE alignments based on the results of these two models as our baseline systems. For the alignments of IBM Model 4 and HMM, we use the published software package, GIZA++³ (Och and Ney, 2003) for processing.

Some recent research has proposed to extract phrase translations based on the results from IBM Model (Koehn et al., 2003). We extract English-Chinese NE alignments based on the results from IBM Model 4 and HMM. The extraction strategy takes each of the continuous aligned segments as one possible candidate, and finally the one with the highest frequency in the whole corpus wins.

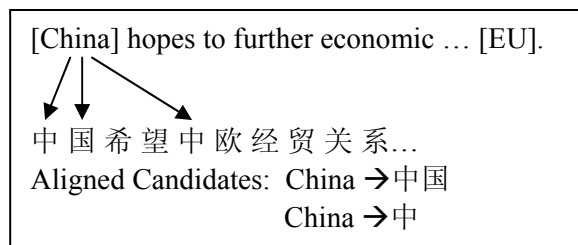


Figure 3. Example of Extraction Strategy

Figure 3 gives an example of the extraction strategy. “China” here is aligned to either “中国” or “中”. Finally the one with a higher frequency in the whole corpus, say, “中国”, will be viewed as the final alignment for “China”.

4.3 Results Analysis

Our approach first uses NLPWIN to conduct NER. Suppose S' is the set of identified NE with NLPWIN. S is the alignment set we compute with our models based on S' , and T is the set consisting of all the true alignments based on S' . We define the evaluation metrics of precision, recall, and F-score as follows:

² <http://www.isi.edu/~och/YASMET.html>

³ <http://www.isi.edu/~och/GIZA++.html>

$$precision = \frac{|S \cap T|}{|S|} \quad (4.1)$$

$$recall = \frac{|S \cap T|}{|T|} \quad (4.2)$$

$$F-score = \frac{2 \times precision \times recall}{precision + recall} \quad (4.3)$$

4.3.1 Results without Word Segmentation

Based on the testing strategies discussed in Section 4.1, we perform all the experiments on data without word segmentation and get the performance for NE alignment with IBM Model 4, the HMM model, and the maximum entropy model. Figure 4, 5, and 6 give the learning curves for precision, recall, and F-score, respectively, with these experiments.

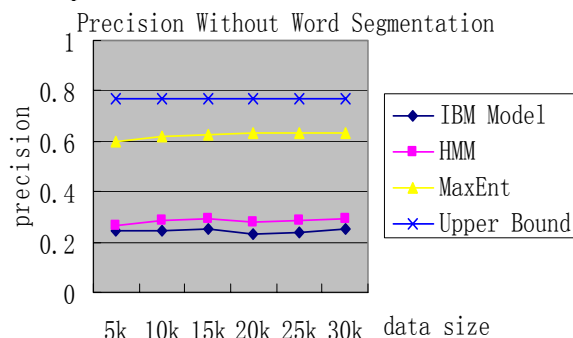


Figure 4. Learning Curve with Precision

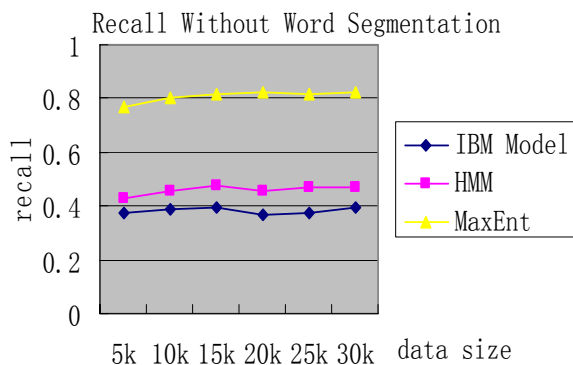


Figure 5. Learning Curve with Recall

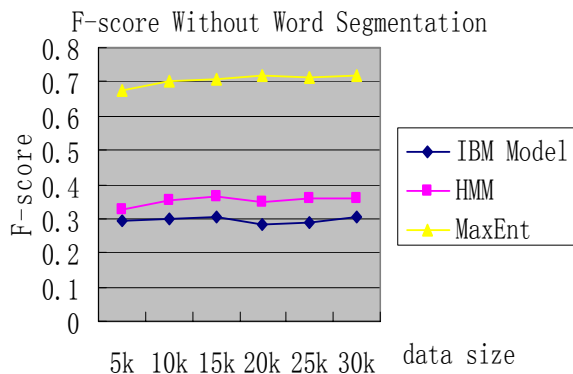


Figure 6. Learning Curve with F-score

From these curves, we see that HMM generally works a little better than IBM Model 4, both for precision and for recall. NE alignment with the maximum entropy model greatly outperforms IBM Model 4 and HMM in precision, recall, and F-Score. Since with this framework, we first use NLPWIN to recognize NEs in English, we have NE identification error. The precision of NLPWIN on our task is about 77%. Taking this into account, we know our precision score has actually been reduced by this rate. In Figure 4, this causes the upper bound of precision to be 77%.

4.3.2 Comparison with Results with Word Segmentation

To justify that our approach of NE alignment without word segmentation really reduces the error propagations from word segmentation and thereafter NER, we also perform all the experiments upon the data set with word segmentation. The segmented data is directly taken from published LDC Xinhua News corpus.

	precision	recall	F-score
MaxEnt (Seg)	0.56705	0.734491	0.64
MaxEnt (Unseg)	0.636015	0.823821	0.717838
HMM (Seg)	0.281955	0.372208	0.320856
HMM (Unseg)	0.291859	0.471464	0.360531
IBM 4 (Seg)	0.223062	0.292804	0.253219
IBM 4 (Unseg)	0.251185	0.394541	0.30695

Table 1. Results Comparison

Table 1 gives the comparison of precision, recall, and F-score for the experiments with word segmentation and without word segmentation when the size of the data set is 30k sentences.

For HMM and IBM Model 4, performance without word segmentation is always better than with word segmentation. For maximum entropy model, the scores without word segmentation are always 6 to 9 percent better than those with word segmentation. This owes to the reduction of error propagation from word segmentation and NER.

For example, in the following sentence pair with word segmentation, the English NE “United States” can no longer be correctly aligned to “美国”. Since in the Chinese sentence, the incorrect

segmentation takes “访问美国” as one unit. But if we conduct alignment without word segmentation, “美国” can be correctly aligned.

- Greek Prime Minister Costas Simitis visits [United States].
- 希腊总理希米蒂斯访问美国.

Similar situations exist when HMM and IBM Model 4 are used for NE alignment. When compared with IBM Model 4 and HMM with word segmentation, our approach with word segmentation also has a much better performance than them. This demonstrates that in any case our approach outperforms IBM Model 4 and HMM significantly.

4.3.3 Discussion

Huang et al.'s (2003) approach investigated transliteration cost and translation cost, based on IBM Model 1, and NE tagging cost by an NE identifier. In our approach, we do not have an NE tagging cost. We use a different type of translation and transliteration score, and add a distortion score that is important to distinguish identical NEs in the same sentence.

Experimental results prove that in our approach the selected features that characterize NE translations from English to Chinese help much for NE alignment. The co-occurrence score uses the knowledge from the whole corpus to help NE alignment. And the transliteration score addresses the problem of data sparseness. For example, English person name “Mostafizur Rahman” only appears once in the data set. But with the transliteration score, we get it aligned to the Chinese NE “穆斯塔菲兹拉赫曼” correctly.

Since in ME training we use iterative bootstrapping to help supervised learning, the training data is not completely clean and brings some errors into the final results. But it avoids the acquisition of large annotated training set and the performance is still much better than traditional alignment models. The performance is also impaired by the English NER tool. Another possible reason for alignment errors is the inconsistency of NE translation in English and Chinese. For example, usually only the last name of foreigners is translated into Chinese and the first name is ignored. This brings some trouble for the alignment of person names.

5 Conclusions

Traditional word alignment approaches cannot come up with satisfactory results for Named Entity alignment. In this paper, we propose a novel approach using a maximum entropy model for NE alignment. To ease the training of the MaxEnt

model, bootstrapping is used to help supervised learning. Unlike previous work reported in the literature, our work conducts bilingual Named Entity alignment without word segmentation for Chinese, and its performance is much better than with word segmentation. When compared with IBM and HMM alignment models, experimental results show that our approach outperforms IBM Model 4 and HMM significantly.

Due to the inconsistency of NE translation, some NE pairs can not be aligned correctly. We may need some manually-generated rules to fix this. We also notice that NER performance over the source language can be improved using bilingual knowledge. These problems will be investigated in the future.

6 Acknowledgements

Thanks to Hang Li, Changning Huang, Yunbo Cao, and John Chen for their valuable comments on this work. Also thank Kevin Knight for his checking of the English of this paper. Special thanks go to Eduard Hovy for his continuous support and encouragement while the first author was visiting MSRA.

References

- Al-Onaizan, Y. and Knight, K. 2002. Translating Named Entities Using Monolingual and Bilingual Resources. *ACL 2002*, pp. 400-408. Philadelphia.
- Berger, A. L.; Della Pietra, S. A.; and Della Pietra, V. J. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, vol. 22, no. 1, pp. 39-68.
- Brew, C. and McKelvie, D. 1996. Word-pair extraction for lexicography. *The 2nd International Conference on New Methods in Language Processing*, pp. 45-55. Ankara.
- Brown, P. F.; Della Pietra, S. A.; Della Pietra, V. J. ;and Mercer, R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263-311.
- Cherry, C. and Lin, D. 2003. A Probability Model to Improve Word Alignment. *ACL 2003*. Sapporo, Japan.
- Darroch, J. N. and Ratcliff, D. 1972. Generalized Iterative Scaling for Log-linear Models. *Annals of Mathematical Statistics*, 43:1470-1480.
- Heidorn, G. 2000. Intelligent Writing Assistant. *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker.

- Hobbs, J. et al. 1996. *FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural Language Text*, MIT Press. Cambridge, MA.
- Huang, F.; Vogel, S. and Waibel, A. 2003. Automatic Extraction of Named Entity Translingual Equivalence Based on Multi-Feature Cost Minimization. *ACL 2003 Workshop on Multilingual and Mixed-language NER*. Sapporo, Japan.
- Huang, J. and Choi, K. 2000. Chinese-Korean Word Alignment Based on Linguistic Comparison. *ACL-2000*. Hongkong.
- Ker, S. J. and Chang, J. S. 1997. A Class-based Approach to Word Alignment. *Computational Linguistics*, 23(2):313-343.
- Knight, K. and Graehl, J. 1997. Machine Transliteration. *ACL 1997*, pp. 128-135.
- Koehn, P.; Och, F. J. and Marcu, D. 2003. Statistical Phrase-Based Translation. *HLT/NAACL 2003*. Edmonton, Canada.
- Lee, C. and Chang, J. S. 2003. Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts, *HLT-NAACL 2003 Workshop on Data Driven MT*, pp. 96-103.
- Marcu, D. 2001. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. *ACL 2001*, pp. 378-385. Toulouse, France.
- Melamed, I. D. 2000. Models of Translation Equivalence among Words. *Computational Linguistics*, 26(2): 221-249.
- Moore, R. C. 2003. Learning Translations of Named-Entity Phrases from Parallel Corpora. *EACL-2003*. Budapest, Hungary.
- Och, F. J. and Ney, H. 2003. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, volume 29, number 1, pp. 19-51.
- Och, F. J. and Ney, H. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *ACL 2002*, pp. 295-302.
- Och, F. J. and Ney, H. 2000. Improved Statistical Alignment Models. *ACL 2000*, pp: 440-447.
- Probst, K. and Brown, R. 2002. Using Similarity Scoring to Improve the Bilingual Dictionary for Word Alignment. *ACL-2002*, pp: 409-416.
- Vogel, S.; Ney, H. and Tillmann, C. 1996. HMM-Based Word Alignment in Statistical Translation. *COLING'96*, pp. 836-841.
- Wang, W.; Zhou, M.; Huang, J. and Huang, C. 2002. Structural Alignment using Bilingual Chunking. *COLING-2002*.