Proceedings of the ACL 2003 Workshop on Multilingual and
Mixed-language Named Entity Recognition, pp. 9-16.

Automatic Extraction of Named Entity Translingual Equivalence Based on Multi-Feature Cost Minimization

Fei Huang, Stephan Vogel and Alex Waibel
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{fhuang, vogel, ahw}@cs.cmu.edu

Abstract

Translingual equivalence refers to the relationship between expressions of the same meaning from different languages. Identifying translingual equivalence of named entities (NE) can significantly contribute to multilingual natural language processing, such as crosslingual information retrieval, crosslingual information extraction and statistical machine translation. In this paper we present an integrated approach to extract NE translingual equivalence from a parallel Chinese-English corpus.

Starting from a bilingual corpus where NEs are automatically tagged for each language, NE pairs are aligned in order to minimize the overall multi-feature alignment cost. An NE transliteration model is presented and iteratively trained using named entity pairs extracted from a bilingual dictionary. The transliteration cost, combined with the named entity tagging cost and word-based translation cost, constitute the multi-feature alignment cost. These features are derived from several information sources using unsupervised and partly supervised methods. A greedy search algorithm is applied to minimize the alignment cost. Experiments show that the proposed approach extracts NE translingual equivalence with 81% F-score and improves the translation score from 7.68 to 7.74.

1. Introduction

Translingual equivalence refers to the relationship between expressions of the same meaning from different languages. Identifying translingual equivalence of named entities (NE), including proper names, temporal and numerical expressions, is very important to multilingual language processing. This is because named entities, especially names of persons, locations and organizations, convey essential meaning in human languages [1][2]. Some approaches for named entity translation, like bilingual dictionary lookup, word/sub-word translation or transliteration, have been explored in the past few years [3][4][5][6][7]. However, dictionary lookup is particularly difficult for translating uncommon NEs because of its limited coverage, and simply applying word-based or character-based transformation without considering their context information, in most cases, cannot achieve satisfactory performance either. For instance, "风 陵 渡/Fenglingdu", a Chinese location name, cannot be found in an LDC dictionary with 50k entries, and it is also inappropriate to adopt the character-by-character translation, "wind tomb cross". Rule-based translation is suitable for temporal and numerical NEs, because of their limited vocabulary and regular usage, but does not generalize well for proper name translation, especially the translation of foreign location or person names.

One possible solution is to automatically extract named entity translingual equivalence from a parallel corpus, where named entities have been manually or automatically annotated. For example, in the following sentence pair where NEs are automatically tagged,

PER{李鹏} 出席 LOC{亚 欧} 会议 后 返 抵 LOC{北京}.
PER{Li Peng} back in LOC{Beijing} after LOC{Attending Asian} LOC{Europe} Meeting .

correct NE alignment requires models for both phonetic transliteration ("*李鹏*" vs. "*Li Peng*") and semantic translation ("*亚 欧 会议" vs. "Asian Europe Meeting"*). Additionally, tagging errors should also be handled. Therefore more sophisticated models that are able to incorporate multiple informative features are necessary.

In this paper we propose an integrated approach to the automatic extraction of named entity translation equivalence from a parallel Chinese/English corpus. Initially, named entities are automatically tagged for each language, after that NE pairs are aligned based on a multi-feature alignment cost minimization strategy. We present a named entity transliteration model and iteratively extract named entities from a bilingual dictionary to train the model. The NE transliteration cost, combined with the NE tagging cost and word-based

translation cost, constitute the multi-feature alignment cost. These features are derived from several information sources using unsupervised and partly supervised methods. A greedy search algorithm is applied to minimize the total alignment cost for each sentence pair. Experiments show that the proposed approach can extract NE translingual equivalence with 81% F-score and improved the translation score from 7.68 to 7.74, which is statistically significant.

The structure of this paper is as follows: in section 2 we introduce the NE transliteration model, in section 3 we propose the multi-feature named entity alignment framework, which incorporates transliteration cost, word-based translation cost and tagging cost. In section 4 we present the experiments and analysis of the results. Conclusions will be given in the last section.

2. Named Entity Transliteration Model

Transliteration is the process of translating certain words (e.g., person's name, location's name) from the source language into their phonetic approximations in the target language. It is an essential component for NE translation. NEs are usually translated by combining the phonetic transliteration of some units and semantic translations of other units, where units can be characters, sub-words or words. Previous work on transliteration ([3],[6]) *explicitly* resorts to phoneme similarities, where a pronunciation lexicon is often needed. Here we try to take the transliteration on the surface level using DP-based string matching.

Directly transliterating Chinese characters into English letters needs a large amount of bilingual NE pairs for training, considering the parameter estimation for over 6,000 frequent Chinese characters. However, intermediate transliteration through pinyin syllables (pinyin is the romanized representation of Chinese characters) is more accurate and easier, because the much smaller alphabet size of pinyin alleviates the data sparseness. Furthermore, pinyin and English letters share a quite similar alphabet that enables the Dynamic Programming (DP)-based string matching.

Mapping Chinese characters to their pinyin syllables (e.g., "萨拉热窝" to "sa la re wo") can be greatly facilitated by a character-pinyin mapping table, which is easy to obtain. However, mapping pinyin syllables to English string (e.g., "sa la re wo" to "Sarajevo") needs more sophisticated models, which usually require a bilingual NE list for training. To acquire an NE list, we propose an unsupervised learning approach in which NE pairs are automatically extracted from a large bilingual dictionary. DP-based string matching is iteratively applied in order to estimate the transliteration probability from pinyin to English letter sequences.

2.1 Transliteration Model Definition

To extract NE pairs from a given bilingual dictionary D, we want to find Chinese-English NE pair (f_{ne}^*, e_{ne}^*) with highest joint probability,

$$\begin{aligned}(f_{ne}^*, e_{ne}^*) &= \arg\max_{(f,e)\subset D} P_{ne}(f,e) \\ &= \arg\max_{(f,e)\subset D} P_{ne}(f) P_{ne}(e \mid f),\end{aligned} \quad (1)$$

where f is the Chinese character sequence and e is the English word string, $P_{ne}(f)$ is the probability of generating the character sequence of the Chinese NE, and $P_{ne}(e \mid f)$ is the probability of *transliterating* the Chinese NE into an English NE.

Suppose f has m characters. For $i = 1,2,..m$, character f_i is transliterated into an English letter string e_i through the pinyin syllable y_i. These strings are non-overlapping. The generation process can be depicted as $f_i \in f \rightarrow y_i \rightarrow e_i \in e$. Here the subscript i indicates that the sub-string is transliterated from f_i, and it is not necessarily the ith word/letter in e.

Let's assume each Chinese character is independently transliterated into an English letter string through its pinyin syllable. Considering that mappings from Chinese characters to their pinyin syllables are mostly deterministic, i.e., $p(y_i \mid f_i) \approx 1$, then

$$P_{ne}(e \mid f) = \prod_{i=1}^{m} p(e_i \mid f_i) = \prod_{i=1}^{m} p(e_i \mid y_i) p(y_i \mid f_i). \quad (2)$$

Suppose y_i is composed of m_i letters, for $j = 1,2,...m_i$, letter $y_{i,j}$ is aligned to letter $e_{i,k}$, where the alignment is represented as $k = a_j$. With the independence assumption,

$$p(e_i \mid y_i) = \prod_{j=1}^{m_i} p(e_{i,k} \mid y_{i,j}) \quad (3)$$

Thus $P_{ne}(e \mid f)$ can be computed as

$$P_{ne}(e \mid f) = \prod_{i=1}^{m} [p(y_i \mid f_i) \prod_{j=1}^{m_i} p(e_{i,k} \mid y_{i,j})] \quad (4)$$

This formula represents two levels of transliteration, Chinese character to pinyin syllable and pinyin syllable to English letter string.

$P_{ne}(f)$ can be computed directly from character language model for Chinese NEs,

$$P_{ne}(f) = p_{ne}(f_1) p_{ne}(f_2 \mid f_1) \prod_{i=3}^{m_e} p_{ne}(f_i \mid f_{i-1}, f_{i-2}) \quad (5)$$

2.2 DP-based Alignment and Iterative Training

Following the derivation of the transliteration model, the next steps are how to identify letter-to-letter alignment and how to train the transliteration model and language model.

Dynamic programming (DP) has been successfully applied in searching for the "optimal" alignment path between two strings, where "optimal" means the minimum accumulated editing cost between aligned word/letter pairs (the cost is usually defined as 0 if they are the same or 1 if there exists insertion, deletion or substitution errors).

Since pinyin and English share a similar alphabet, the DP-based string alignment is also applicable. However, the original binary cost function is not appropriate for pronunciation-based transliteration. Now the phonetic similarity is more important than the orthographic similarity [3], therefore alignment cost between letters with similar pronunciations (e.g., "c" and "k" or "p" and "b") should be smaller.

One alternative is to take the minus logarithm of the letter transliteration probability as the matching cost, i.e., the cost of aligning letter $e_{i,k}$ from English and letter $y_{i,j}$ from pinyin is defined as,

$$C(e_{i,k}, y_{i,j}) = -\log p(e_{i,k} \mid y_{i,j}). \quad (6)$$

This cost function is defined directly from transliteration probabilities. It allows both self-transliteration and the transliteration from letters with similar pronunciations. Thus it is more general and accurate. Further more, the final accumulative alignment cost between pinyin syllables and English words also corresponds to the word/character-level transliteration cost.

To calculate the alignment the transliteration model parameters have to be known, which in turn are computed based on the relevant alignment frequency, i.e.,

$$p(e_{i,k} \mid y_{i,j}) = \frac{C(e_{i,k}, y_{i,j})}{\sum_{e'} C(e', y_{i,j})}, \quad (7)$$

where $C(e_{i,k}, y_{i,j})$ is the frequency that $e_{i,k}$ and $y_{i,j}$ is aligned. To resolve this inter-dependence between models, the original binary cost function is first applied to the DP-based string alignment. A list of bilingual NE pairs is extracted from the dictionary according to their alignment cost. Based on this initial imperfect name list, the letter transliteration model and character language model are trained, and employed for the NE joint probability estimation (see formula (1), (4) and (5)). In the following iterations, the alignment cost function as well as the transliteration probability is updated, NE pairs are selected according to their joint probabilities, and translation and language models are re-trained using the cleaner NE list. Experiment results in section 4 show that an unsupervised learning approach improves the accuracy of extracted NE list by refining both translation and language models iteratively.

3. Multi-Feature Named Entity Alignment Model

To align the NE translingual equivalence within a sentence pair, we adopt the NE alignment model which incorporates several features for cost estimation and minimizes the total cost for the given pair. These features include NE transliteration cost, word-based NE translation cost and NE tagging cost, and will be discussed in more details in the following sections.

3.1. Named Entity Transliteration Cost

The translation of different NE equivalences is highly type-dependent. While most PERSON and LOCATION NE equivalences can be transformed primarily through transliteration, some LOCATION and most ORGANIZATION NE equivalences are transformed by combining both semantic translation and phonetic transliteration. For example, translating a location name "萨拉热窝 机场" needs both phonetic transliteration of the specific city name ("萨拉热窝" to "Sarajevo") and semantic translation of the general facility ("机场" to "airport").

To deal with this problem, we adopted a translation-based transliteration approach, similar to the candidate generation approach proposed by [4],. For each word in the Chinese NE candidate, its transliteration could be either pinyin or its semantic translation(s) from the bilingual dictionary, and can be aligned to any word in the English NE candidate. By way of a greedy search algorithm (detailed discussion in section 3.4), each English word is aligned to a unique Chinese word such that their transliteration cost is the minimum among the unaligned word pairs. The total NE transliteration cost is the sum of the word-to-word transliteration costs along the alignment path.

Let A^* denotes the "*optimal*" alignment path, and let f_i, e_j be the i th word in Chinese NE f_{ne} and the j th word in English NE e_{ne}, respectively. Then

$$\begin{aligned} C_{translit}(f_{ne}, e_{ne}) &\equiv C_{translit}(f_{ne}, e_{ne} \mid A^*) \\ &= \sum_{(i,j)\in A^*} C_{translit}(f_i, e_j) \\ &= \sum_{(i,j)\in A^*} [\arg\min_{\{y_i \in E_{fi}\}} - \log P(e_j \mid y_i)], \end{aligned} \quad (8)$$

where y_i is one element in f_i's transliteration candidate set E_{f_i}.

This approach allows the alignment between any word pairs, so it is not sensitive to the word orders in NE pairs, and therefore can handle flexible combination of translation and transliteration. It is also robust to inflectional forms (e.g., the plural form of nouns) in English NEs.

3.2. Named Entity Translation Cost

Word translation probabilities can be estimated from a parallel corpus using various well-known alignment models, such as the IBM-model and HMM-model [8][9]. They can be further used to calculate the probability that a Chinese NE is the translation of an English NE on the word level.

Assume the English NE e_{ne} has L English words, $e_1, e_2, ...e_L$, and the Chinese NE f_{ne} has J Chinese words, $f_1, f_2, ...f_J$. The translation probability of the named entities pair is computed using the IBM model-1, as:

$$P_{trans}(f_{ne} \mid e_{ne}) = \frac{1}{L^J} \prod_{j=1}^{J} \sum_{l=1}^{L} p(f_j \mid e_l) \tag{9}$$

This alignment model is asymmetric, as one source word can only be aligned to one target word, while one target word can be aligned to multiple source words. To make it symmetric, we estimate both $P(f_{ne} \mid e_{ne})$ and $P(e_{ne} \mid f_{ne})$, and define the NE translation cost as:

$$\begin{aligned} & C_{trans}(e_{ne}, f_{ne}) \\ & \equiv C_{trans}(e_{ne} \mid f_{ne}) + C_{trans}(f_{ne} \mid e_{ne}) \\ & \equiv -\lambda_p [\log P_{trans}(e_{ne} \mid f_{ne}) + \log P_{trans}(f_{ne} \mid e_{ne})], \end{aligned} \tag{10}$$

That is, the translation cost of a given NE pair (e_{ne}, f_{ne}) is composed of the sum of the bi-direction translation cost, and weighted by position match weight λ_p.

λ_p models the "distance" between the relative positions of aligned NEs in each sentence. It is characterized by a normal distribution,

$$\lambda_p = \frac{1}{\sqrt{2\pi\sigma}} \exp\{-\frac{(p_f - p_e)^2}{2\sigma^2}\} \tag{11}$$

where p_f, the relative position of a Chinese NE covering words from w_i to w_j, is defined as

$$p_f = (i + j)/2J. \tag{12}$$

The relative position of an English NE, p_e, is defined similarly. The variance σ is empirically chosen according to the homogeneity of word orders between two languages.

3.3. Named Entity Tagging Cost

An NE tagging software, IdentiFinder™, automatically tags NEs for both English and Chinese. When evaluated on the bilingual corpus, the tagging accuracy for ENAMEX type (including PERSON, LOCATION and ORGANIZATION) NEs is about 80%. Those tagging errors, including missing, spurious (false positive) and partial tagging, inevitably introduce errors into NE alignment. It would be helpful to know the confidence or the probability that a tagged word sequence is a real NE. Unfortunately, the outsourced tagger doesn't output such information. To get this information, an HMM-based NE tagger is trained from the *imperfect* training corpus, i.e., the automatically tagged bilingual corpus containing incorrect NEs.

Automatic named entity tagging based on HMM has achieved satisfactory performance [2]. In this framework, each type of NEs as well as the remaining type, NOT_A_NAME, is represented by a unique internal state in the HMM. An NE-tagged sentence is generated according to the following assumption:

1. The current NE type is selected according to the previous word and its NE type, with type transitional probability $P_c(N \mid w_{-1}, N_{-1})$;
2. The first word in a NE is generated according to the current and previous NE types, with first word generation probability $P_f(w_1 \mid N, N_{-1})$;
3. Each subsequent word in this NE is generated from a type-dependent bigram model, with probability $P_b(w \mid w_{-1}, N)$.

In the above notation, N and N_{-1} represent the current and previous NE type respectively, w_1 represents the first word in the current NE type, w represents the current word, and w_{-1} represents the previous word.

Given a sequence of words $\overrightarrow{W} = (w_1, w_2, ..., w_n)$ and its corresponding NE type sequence $\overrightarrow{N} = (N_1, N_2, ..., N_n)$, the probability of generating the words from the NE type sequence is defined as

$$P(\overrightarrow{W} \mid \overrightarrow{N}) = \prod_{i-1}^{n} p(w_i, N_i \mid w_{i-1}, N_{i-1}) \tag{13}$$

where $p(w_i, N_i \mid w_{i-1}, N_{i-1})$ denotes the transitional probability from w_{i-1} to w_i, given that the corresponding NE types are N_i and N_{i-1}. When the transition is within the same NE, i.e., $N_i = N_{i-1}$, this is

just the type-dependent bigram model $P_b(w_i \mid w_{i-1}, N_i)$. When the transition is between different NEs, this becomes the product of type transition probability $P_c(N_i \mid w_{i-1}, N_{i-1})$ and first word generation probability $P_f(w_i \mid N_i, N_{i-1})$.

For an aligned NE pair (e_{ne}, f_{ne}), their NE types should be the same. So the NE tagging cost is defined as

$$C_{tag}(e_{ne}, f_{ne}) = \min_N [-\log P(e_{ne} \mid N) - \log P(f_{ne} \mid N)] \quad (14)$$

This criteria chooses the best NE type N from PERSON, LOCATION and ORGANIZATION that generates the Chinese and English NE word sequences with the highest probabilities. During parameter estimation, to reduce negative effect from erroneous initial tagging, the corpus is split into 2 parts, and the model is trained from one half when applied on the other half.

3.4 Multi-Feature Cost Minimization

Provided with different alignment features, including the transliteration cost $C_{translit}$, the translation cost C_{trans} and the tagging cost C_{tag}, the overall alignment cost for the NE pair (e_{ne}, f_{ne}) is their linear combination:

$$C(e_{ne}, f_{ne}) = \lambda_1 C_{translit}(e_{ne}, f_{ne}) + \lambda_2 C_{trans}(e_{ne}, f_{ne}) + \lambda_3 C_{tag}(e_{ne}, f_{ne}) \quad (15)$$

where the three λ's are either empirically chosen to discriminate correct and incorrect NE alignments with best accuracy, or selected according to the quality/confidence of each feature model. In the current implementation, these parameters are selected to map the three weighted costs into the same numerical range while putting a little less confidence on C_{tag} (since it is trained from imperfect training data).

For any bilingual sentence pair containing multiple NEs, the desirable alignment scheme should minimize the sum of the overall alignment cost. To find this optimal alignment, an algorithm similar to the competitive linking algorithm [10] is adopted:

1. Initialize *NE-Aligned* to be an empty set and *NE-Pairs* as the list of all possible combinations of a source language NE and a target language NE in the given sentence pair;
2. Sort *NE-Pairs* in ascending order according to their overall alignment cost defined in Formula (15);
3. Move the topmost pair (e_{ne}, f_{ne}), i.e. the pair with the smallest alignment cost from *NE-Pairs* to *NE-Aligned*;
4. Remove all entries containing either e_{ne} or f_{ne} from *NE-Pairs*, with the assumption that once a Chinese NE is aligned with an English NE, it can't be aligned with any other English NE. The same is true for English NEs;
5. Repeat from Step 3 until *NE-Pairs* is empty or the top alignment cost is above a certain threshold. The resultant *NE-Aligned* leads to the "optimal" alignment.

Note that this algorithm is based on a greedy search approximation, i.e., it only chooses the local optimal alignment—the currently minimum cost alignment pair among unaligned pairs—at each step, it cannot guarantee the global optimality. But empirically it often finds the alignment with minimum or close to minimum sentence alignment cost.

3.5 Open-End NE Alignment

When applying extracted NE equivalences to the statistical machine translation task (see section 4.3), the translation score is improved. Detailed analysis shows that initial tagging errors still cause many problems for NE translation. Some NEs in the test data are not translated correctly because they are untagged, partially tagged or tagged with other words as one NE in the training corpus. For example,

记者 ***PERSON***{王} 能 标 ***ORGANIZATION***{赞比亚 司法部} 长 ***PERSON***{马兰} 博 12 日在 此间 会见 ***LOCATION***{中国} 监察 代表团

should be tagged as

记者 ***PERSON***{王 能 标} ***LOCATION***{赞比亚} 司法部 长 ***PERSON***{马 兰 博} 12 日在 此间 会见 ***ORGANIZATION***{中国 监察 代表团}

To recover from those partial tagging errors, an open-end NE alignment window is utilized. The window is initially set to fit the originally tagged NEs, afterwards both ends of the window are allowed to expand and shrink within a given range. As a result, optimal aligned NEs are searched from all word sequences within the resultant variable-length sliding window.

4. Experiment Result and Discussion

4.1 Named Entity Transliteration

Three sets of experiments are conducted. The first one is to evaluate the proposed iterative training for the NE transliteration model by examining the accuracy of the extracted NE lists. The second is to measure the precision/recall of NE pair extracted from a small data set, and the third is to assess the increased translation quality by adding the NE bilingual dictionary. The

bilingual dictionary used to train the transliteration model is the Chinese-English dictionary version 3.1 released by the Linguistic Data Consortium (LDC). This dictionary contains 81,945 entries for 54,131 unique Chinese words.

Initially we extracted 3,000 NE pairs with minimum string matching cost under a 0/1 cost function. From this small name list, the letter transliteration model and Chinese character language model are trained and integrated into the statistical transliteration framework. In the following extraction iteration, additional 500 named entity pairs with higher NE joint probabilities are added and used to update the transliteration model and the language model. This process continues until adding more NE pairs doesn't improve the extraction accuracy any more, which usually happens at the 6th iteration where a total of 5,500~6,000 NE entries are included.

Because NE pairs are sorted descendingly according to their joint probabilities, entries at the top of the sorted list are more likely to be NE pairs than those at the bottom. To estimate the overall accuracy for all extracted NEs, we evaluate the "local" accuracy of evenly distributed segments in the sorted NE pair list. In other word we count the number of correct NE pairs for each segment located at the 0-100th, 900th-1000th, 1900th–2000th NE pair, etc.. The precision evaluated on these sub-samples is used to estimate the overall accuracy.

Figure 1 shows the precision curve after selected iterations at different evaluation segments. "0/1 baseline" represents the result when using only DP string matching with the 0/1 cost function. "Ite*x*" means the result after the *x*th iteration. One can see that for well-trained models (after the 4th iteration) the accuracy of the evaluation segment at 5000th just slightly degrades compared with those top segments. The precisions of all the segments are consistently increased after each iteration. One can see that the most significant accuracy degradation happens at the 6000th segment. This indicates that most NE pairs in the dictionary have already been included, and adding more non-NE entries will "pollute" the transliteration model, thus the performance can become even worse.

4.2 Extracting Named Entity Translingual Equivalence

The bilingual corpus contains sentence-aligned newswire data from the Xinhua News Agency and the Foreign Broadcast Information Service (FBIS). Some bilingual sentence pairs are automatically extracted and aligned, therefore there exist errors in both alignment and translation. The Chinese sentences are pre-segmented using a maximum-matching segmenter with a 44K wordlist. Totally there are 152,391 sentence pairs, about 6 million English words and 5.5 million Chinese words. Named entities in the bilingual corpus are first annotated using BBN's IdentiFinder™, then aligned according to the multi-feature cost minimization framework.

For the purpose of evaluation, a small set of test data is randomly selected, which contains 100 sentence pairs, 4950 Chinese words and 5646 English words. The number of named entity pairs which can be aligned is 357. These named entities are manually annotated and aligned, and used as the gold standard to evaluate the automatically extracted and aligned NE pairs.

Table 1 shows the precision/recall/F-score using different feature sets for cost minimization. "C_{trans}" means using word-level translation cost only, "+C_{tag}" means adding NE tagging cost, "+ $C_{translit}$" means adding NE transliteration cost into the previous feature set. It can be seen that by adding more information, both precision and recall are improved. Tagging cost and transliteration cost individually lead to about 3% increase in F-score and the overall improvement is about 6.8%. The last row shows the NE alignment accuracy on manually annotated test data, where all tagging errors have been corrected. The significant improvement in F-score (81.3% to 93.7%) indicates that initial tagging errors remain the major cause of alignment errors.

Figure 2 demonstrates some examples of extracted NE translation equivalences from the given sentence pairs, when applying various models. In each example NE pairs with the same number label (e.g., C1 and E1) are considered correct alignment. One can see from example 1 that the proposed alignment strategy can correctly align most NE pairs, even with NE translation cost only. Those incorrect alignments, marked by (*), are caused either by missed NE tagging or non-exact translations. When adding tagging cost, some missed NEs could be recovered and correctly aligned (See example 2). Example 3 shows that the transliteration model works best for NEs containing people's name.

4.3 Improving Translation Quality with Named Entity Dictionary

A NE translation dictionary can be constructed from extracted NE equivalences. In the dictionary one Chinese NE may have multiple English translations with different probabilities. These probabilities are proportional to the frequencies of the NE alignments. This dictionary is integrated into a statistical machine translation (SMT) engine and evaluated on Chinese-English newswire translation.

The SMT system is based on weighted finite state transducers [11], where each transducer is a collection of bilingual equivalence for words, phrases or NEs. In our experiment, three transducers are integrated into the translation engine,

- A word level transducer, which is essentially from the LDC Chinese-English dictionaries (see Section 4.1). Since many entries in this dictionary are manually compiled, this dictionary has very high accuracy. It is called "LDC" in table 2.
- A phrase-to-phrase transducers where the phrase pairs are extracted from the HMM Viterbi alignment [9] for each sentence pair in the bilingual corpus. It is called "HMM" in table 2.
- A NE transducer from the NE translation dictionary, the "NE" in table 2.

The evaluation data is the same newswire data used in TIDES 2001 dry-run evaluation. It contains 993 sentences, 24,821 words. From this data set the IdentiFinder™ extracted 2,379 NEs with totally 3,597 words. Evaluation metrics are fully automatic, including Bleu and NIST8 scores. Table 2 shows the improvement on translation score after adding the NE transducer to various transducer settings. From the table we can see that the NE transducer gives statistically significant improvement in all the settings, although the amount of improvement varies from 0.06 to 0.45. This is because there are some overlaps between the NE transducer and the HMM phrase transducer.

5. Conclusion

We proposed an approach to the automatic extraction of named entity translation equivalence from a parallel Chinese/English corpus based on multi-feature cost minimization. We presented a named entity transliteration model and iteratively extracted named entities from a bilingual dictionary to train the model. The NE transliteration cost, the NE tagging cost and word-based translation cost constitute the multi-feature alignment cost. These features are derived from several information sources using unsupervised and partly supervised methods. Experiments showed that the proposed approach can extract NE translingual equivalence with 81% in terms of F-score and significantly improved the translation score from 7.68 to 7.74.

6. Acknowledgement

We cordially thank the anonymous reviewers for their valuable comments and suggestions in order to prepare the final version of this paper. We also thank BBN for providing us with their named entity tagging software IdentiFinder™.

Table 1. Precision/Recall of Extracted NE Translingual Equivalence

	Precision	Recall	F-score
C_{trans}	66.1%	85.5%	74.5%
$+C_{tag}$	69.7%	87.7%	77.7%
$+C_{translit}$	73.8%	90.5%	81.3%
Manual Annotation	91.3%	96.1%	93.7%

Table 2. Translation Quality Improvement by Adding NE Dictionaries

	Bleu Score	NIST8 Score	t-test statistics
LDC	0.131	6.193	t=8.516
LDC+NE	0.151	6.644	p=0.000
LDC+HMM	0.210	7.677	t=1.963
LDC+HMM+NE	0.213	7.744	p=0.026

7. References

[1] D. Appelt, J. Hobbs, D. Israel, and M. Tyson. Fastus: A finite-state processor for information extraction from real world texts. *In Proceedings of IJCAI-93*, pp.1172-1178, Chambery, France, 1993.

[2] D. Bikel, S. Miller, R. Schwarz and R. Weischedel. Nymble: A high-performance learning name-finder. *In Proceedings of Applied Natural Language Processing,* pp.194-201, Washington DC, 1997.

[3] K. Knight and J. Graehl. Machine Transliteration. *In Proceedings of the ACL '97*. pp.128-135, Somerset, New Jersey, 1997.

[4] Y. Al-Onaizan and K. Knight. Translating Named Entity Using Bilingual and Monolingual Resources, *in Proceedings of Association of Computational Linguistics 2002*, Philadelphia, PA, July, 2002.

[5] B. Stalls and K. Knight. Translating Names and Technical Terms in Arabic Text. *In Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*, 1998.

[6] H. Meng, W. K. Lo, B. Chen and K. Tang. Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval. *In Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Trento, Italy, December 2001.

[7] F. Huang and S. Vogel. Improved Named Entity Translation and Bilingual Named Entity Extraction, *In*

Proceedings of the 4th IEEE International Conference on Multimodal Interface, pp. 253-258, Pittsburgh, PA, October 2002.

[8] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R.L. Mercer. The mathematics of Machine Translation: Parameter Estimation. *In Computational Linguistics,* vol 19, number 2. pp.263-311, June, 1993.

[9] S. Vogel, H. Ney and C. Tillmann. HMM-Based Word Alignment in Statistical Translation. *In Proceedings of the ACL'96*, pp. 836-841, Copenhagen, Denmark, August 1996.

[10] I. Dan Melamed. Models of Translational Equivalence among Words, *In Computational Linguistics 26(2),* pp. 221-249, June 2000.

[11] S. Vogel and H. Ney. Translation with Cascaded Finite State Transducers. *In Proceedings of the ACL'00*, pp. 23-30, Hong Kong, China, October, 2000.

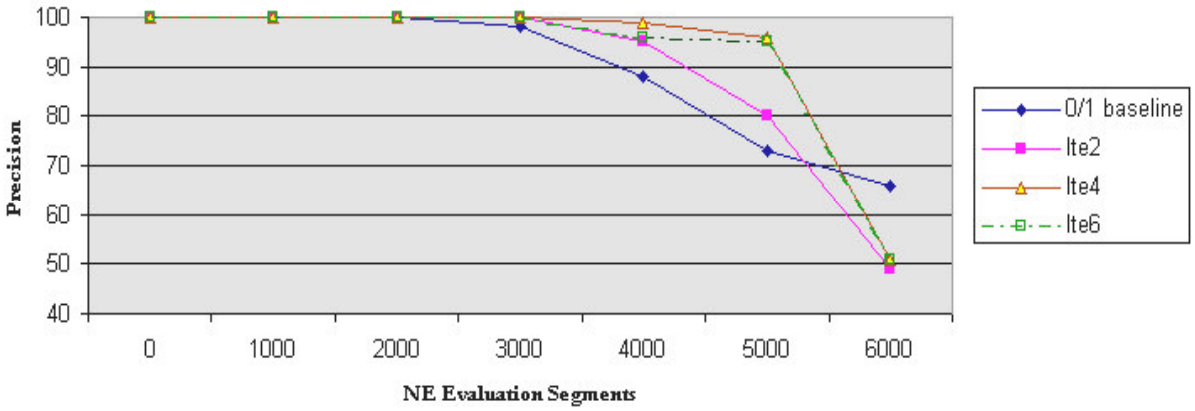

Figure 1. Precision Curve of Evaluation Segments

Example1: 对 LOC_C1{德国} 来 说 , 重要的 LOC_C2{亚洲} 市场 是 LOC_C3{日本} , LOC_C4{中国} , LOC_C5{韩国} , LOC_C6{台湾} , LOC_C7{香港} 以及 LOC_C8{东盟} 等 国家 和地区 , 其中 尤 以 向 LOC_C9{东盟} 国家 的 出口 增长 最 快 .
It noted that LOC_E3{Japan} , LOC_E4{China} , LOC_E5{South Korea} , LOC_E6a{China}'s LOC_E6b{Taiwan} , LOC_E7{Hong Kong} and member states of the ORG_E8a{Association of Southeast Asian Nations} (ORG_E8b{ASEAN}) are all major markets for German products , adding that German exports to ORG_E9{ASEAN} countries increased faster than in other markets . In 1994 , for example , German exports to LOC_E10{Malaysia} increased by 41 percent and to LOC_E11{Thailand} by 33.5 percent.

C_{trans}

LOC_C7{香港}	-----	LOC_E7{Hong Kong}	LOC_C5{韩国}	-----	LOC_E5{South Korea}
LOC_C6{台湾}	-----	LOC_E6b{Taiwan}	LOC_C3{日本}	-----	LOC_E3{Japan}
LOC_C4{中国}	-----	LOC_E4{China}	LOC_C9{东盟}	-----	ORG_E9{ASEAN}
LOC_C8{东盟}	-----	ORG_E8a{Association of Southeast Asian Nations}			
(*)LOC_C2{亚洲}	-----	LOC_E6a{China}	(*)LOC_C1{德国}	----	LOC_E11{Thailand}

Example2: ORG_C1{中 葡 联合 联络 小组} 第十 五次 全体会议 将于 今年 11 月 10 日 至 13 日 在 LOC_C2{北京} 举行 。
the 15th plenary session of the sino-portuguese joint liaison group is scheduled to be held between november 10 and 13 this year in LOC_E2{beijing}.

C_{trans} LOC_C2{北京} ----- LOC_E2{beijing}

$C_{trans} + C_{tag}$ LOC_C2{北京} ----- LOC_E2{beijing}

ORG_C1{中 葡 联合 联络 小组} ----- ORG_{sino-portuguese joint liaison group}

Example3: ORG_C1{外交部} 发言人 PER_C2{章} 启 月 今天 在 此间 宣布
ORG_E1{foreign ministry} spokeswoman PER_E2{zhang qiyue} announced here today

$C_{trans} + C_{tag}$ ORG_C1{外交部}	-----	ORG_E1{foreign ministry}	PER_C2{章}	-----	PER_E2{zhang qiyue}
$C_{trans} + C_{tag} + C_{translit}$ ORG_C1{外交部}	-----	ORG_E1{foreign ministry}	PER_C2{章 启 月}	-----	PER_E2{zhang qiyue}

Figure 2. Selected Parallel Sentences and extracted NE equivalences from Different Feature Combination