Improving Named Entity Translation Combining Phonetic and Semantic Similarities

Fei Huang, Stephan Vogel and Alex Waibel
Language Technologies Institute
School of Computer Sciences
Carnegie Mellon University
{fhuang,vogel,ahw}@cs.cmu.edu

Abstract

This paper describes an approach to translate rarely occurring named entities (NE) by combining phonetic and semantic similarities. The phonetic similarity is estimated from a surface string transliteration model, and the semantic similarity is calculated from a context vector semantic model. Given a source (Chinese) NE and its context, this approach first generates queries in the target (English) language according to the context translation hypotheses, then searches for relevant documents from a target language corpus. Target NEs in retrieved documents are compared with the source NE based on their phonetic and contextual semantic similarities, and the best-matched one is selected as the correct translation. Experiments show that this approach achieves 67% accuracy on translating rarely occurring NEs, and consistently improves the translation quality on different tasks over a state-of-the-art statistical machine translation system.

1 Introduction

Translating Named Entities (NE), in particular named persons, locations and organizations, can benefit many natural language processing tasks. Correct NE translations often act as either key queries in cross-lingual information retrieval or correct answers in multilingual question answering. Moreover, in machine translation, incorrect NE translations not only discard meaningful information from the original sentences, but also introduce a distorted context which degrades the overall translation quality. However, translating NEs is also a challenging problem. Part of the reason is that NEs are either phonetically transliterated (mostly for person names) or semantically translated (mostly for organization names) or both (mostly for location names, like "Appalachian Mountains"), and often there is no one-to-one mapping in transliteration and translation between source and target languages. Although pre-compiled NE translation dictionaries may help in translating some frequent NEs, such as the names of countries, big companies or famous persons, it cannot handle the translation of rarely occurring names, especially new names. For example, in the 2001 Chinese-English translation evaluation test data, 20% of the automatically tagged Chinese NEs are not included in the 50K LDC Chinese-English translation lexicon.

Although many research efforts have been focused on automatic NE detection, and good performance has been achieved in some languages (Chinchor 1997), there are still many areas in NE translation that call for further investigation. (Knight and Graehl 1997) proposed a generative model for Japanese-English back transliteration, (Stalls and Knight 1998) expanded that model to Arabic-English transliteration, and (Al-Onaizan and Knight 2002) additionally incorporated web counts to re-score the transliteration candidates. (Meng et al. 2001) developed an English-Chinese NE transliteration technique using a pronunciation lexicon and phonetic mapping rules. (Moore 2003) proposed statistical phrase translation models to find NE translations in English-French software manuals. (Huang et al. 2003) extracted NE translation pairs from a Chinese-English parallel corpus combining letter transliteration, word translation and NE tagging features, then constructed an NE translation dictionary based on alignment costs and frequencies.

Aligning NE translations from a parallel corpus usually achieves high accuracy on frequently occurring NEs, but it fails in translating rarely occurring NEs which may not appear in the bilingual corpus, as shown in the following example:

Chn: 荷兰 驻 华 大使 郝 德 扬 先生
Ref.: netherlands' ambassador to china, **van houten**
Hyp: netherlands ambassador **hao germany hurl**s

It is noticed that "van houten", the ambassador's name, was not included in the translation lexicon and parallel corpus, thus the name was inappropriately semantically translated character by character, "郝/hao 德/germany 扬/hurls".

In this paper we will propose an approach focusing on translating these rarely occurring NEs. Given a Chinese NE and its context (e.g., the document where the NE appears), this approach first generates queries in English according to the initial document translation hypotheses, then searches for relevant documents from an English corpus using a search engine. It compares the Chinese NE with English NEs in retrieved documents based on their phonetic and semantic similarities, and selects the best-matched one as the translation. The phonetic similarity is calculated from the surface string transliteration model, and the semantic similarity is measured according to the "distance" between the two NEs' context vectors, where the context vector is constructed based on the part-of-speech (POS) and relative locations of the NEs' surrounding words. Experiments show that NE translation achieves a 67% accuracy with the combined similarity models, and the translation quality is consistently better on different translation tasks than a state-of-the-art statistical machine translation system.

The structure of this paper is as follows: in section 2 we introduce the surface string transliteration model; in section 3 we describe the contextual semantic similarity model; we detail the query generation and retrieval process in section 4. In section 5 we present the experiments and analysis of the results. Conclusions will be given in the last section.

2 Surface String Transliteration Model

NE transliteration is the phonetic translation based on pronunciation similarities between source and target NE pairs. Considering that person and location names are often phonetically translated and their written forms resemble their pronunciations, it is possible to discover NE translation pairs through their written forms, i.e., surface string transliteration. Compared with the traditional phoneme transliteration method, surface string transliteration does not require a pronunciation lexicon, which is an advantage especially for rare names. For non-Latin-derived languages like Chinese and Arabic, indirect surface string transliteration is feasible through a romanization process which maps each character into one or more Latin letters with similar pronunciation. For example, the Chinese word "菲 茨 沃 特" is romanized as the *pinyin* form "fei ci wo te", which is the translation of "fitzwater".

Mappings between Chinese characters and their pinyin forms are usually deterministic, while mappings between pinyin and English letters are more sophisticated, and can be learned from a bilingual NE list. To acquire such an NE list, we propose an unsupervised learning approach in which NE pairs are automatically extracted from a large bilingual dictionary. Dynamic programming (DP)-based string alignment is iteratively applied in order to find NE pairs to estimate the transliteration probability from pinyin to English letter sequences.

To extract the NE pair $(f_{ne}{}^*, e_{ne}{}^*)$ from a given bilingual dictionary D, we want to find the entry with the 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 $P_{ne}(f)$ is the probability of generating the character sequence of the Chinese NE, which can be computed directly from a character language model for Chinese NEs. The estimation of $P_{ne}(e \mid f)$, the probability of *transliterating* the Chinese NE f into an English NE e, is as follows.

Suppose f has m characters. For $i = 1,2,..m$, character f_i is mapped into its pinyin syllable y_i, which is further transliterated into an English letter string e_i. Given that mappings from Chinese characters to their pinyin syllables are mostly deterministic, i.e., $p(y_i \mid f_i) \approx 1$, we have

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

Suppose y_i is composed of m_i letters, and for $j = 1,2,...m_i$, the pinyin letter $y_{i,j}$ is aligned to $e_{i,k}$, the k th letter in e_i, where the alignment is represented as $k = a_j$. Assuming independence of transliterated letters we obtain,

$$P_{ne}(e \mid f) \approx \prod_{i=1}^{m} p(e_i \mid y_i) = \prod_{i=1}^{m} \prod_{j=1}^{m_i} p(e_{i,k} \mid y_{i,j}). \quad (3)$$

That is, the transliteration probability between a Chinese NE and an English NE is approximated by the product of their letter transliteration probabilities.

Dynamic programming has been successfully applied to find the "optimal" alignment path between two strings, where "optimal" means the minimum accumulated editing cost between aligned word/letter pairs (Levenstein 1965). Here the cost is usually defined as 0 if they are the same or 1 in case of an insertion, deletion or substitution error. However, this binary cost function is not appropriate for pronunciation-based transliteration, because the phonetic similarity is more important than the orthographic similarity; therefore, the alignment cost between letters with similar pronunciations (e.g., "c" and "k" or "p" and "b") should be smaller. We take the

negative logarithm of the letter transliteration probability as the matching cost, where the transliteration probabilities are computed based on their alignment frequency. However, the alignment frequency is counted under a certain alignment cost function. To resolve this model interdependency, the binary cost function is initially applied to the DP string alignment. Bilingual NE pairs are 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. In the subsequent iterations, the alignment cost function as well as the transliteration probability is updated, NE pairs are re-selected according to their joint probabilities, and transliteration and language models are re-trained using the cleaner NE list.

3 Contextual Semantic Similarity Model

Surface string transliteration model is effective in finding NE translation pairs with similar pronunciations and spellings, but it is weak at identifying NE pairs with dissimilar pronunciations or discriminating different target NEs with similar pronunciations. On the other hand, NEs often occur within certain semantically related contexts, such as the title of a person or the neighbor area of a location. It is possible to combine the context's semantic similarity with the phonetic similarity to improve the NE translation accuracy. As shown in the previous example, although the pronunciations between "郝/hao 德/de 扬/yang" and "van houten" are less similar, the common context with which they both occur, (here it is the title of the named person, "netherlands' ambassador to china", although expressed in different languages), indicates the strong association between the source NE and the target NE.

Different context words have different power in predicting an NE's meanings; in other words, they have different semantic correlation weights with regard to the NE. The context words and their correlation weights can be represented by a context vector, which characterizes the NE's topical information. In this section, we will describe how to create a context vector for a given NE, and how to calculate the semantic similarity between the source and target context vectors.

3.1 Context Vector Selection

A context vector represents the words within a certain context of a given NE, while each word has a different weight reflecting its semantic significance to the NE. Our task is to select context vector words and calculate their correlation weights based on their POS tags and distances to the NE. For each NE-word pair, the word's correlation to the NE is initially measured by Phi-square coefficients, which are further used for estimating the weights of different POS tags and locations. The POS tag weights imply the types of words that should be included in the context vector, and the location weights indicate the optimal length of the context vector.

While mutual information describes the independence between random variables, Chi-square, including its variant, Phi-square, is better at correlating two categorical variables. Unlike Chi-square, Phi-square's value ranges from 0 (no correlation between the two variables) to 1 (perfect correlation between them), thus a probabilistic interpretation is possible. In our problem, we want to measure the correlation between an NE and its context word, so the NE-word semantic correlation coefficient can be defined as:

$$\phi(n,w)=\frac{(o_{11}o_{22}-o_{12}o_{21})}{\sqrt{(o_{11}+o_{12})(o_{11}+o_{21})(o_{21}+o_{22})(o_{12}+o_{22})}}, \quad (4)$$

where n, w are the NE and its context word respectively, $o_{11}, o_{22}, o_{21}, o_{12}$ are the frequencies that they co-occur, that neither occur, and that one occurs and the other does not occur. The higher the coefficient, the more likely is it that the NE and the word are semantically correlated.

To estimate a POS tag's semantic significance to an NE, we calculate the mean of the correlation weight over all NE-word pairs. The correlation weight is also weighted by the probability that the word's POS is the current POS tag. Suppose under the empirical NE-word pair distribution $f(n,w)$, t is the POS tag of w, which is a context word of an NE n, and then $p(t \mid w)$ is the probability that word w has POS tag t, the POS tag's weight is defined as:

$$\begin{aligned} W(t) &\equiv E_{f(n,w)}[p(t \mid w)\phi(n,w)] \\ &= \frac{1}{\sum_{(n,w)} C(n,w)} \sum_{(n,w)} C(n,w)\, p(t \mid w)\phi(n,w), \end{aligned} \quad (5)$$

where $C(n,w)$ is the frequency that (n,w) co-occur.

Figure 1 illustrates the normalized weights of different English POS tags, where one can observe that high correlations are often associated with content words (e.g., nouns, verbs and adjectives are likely the most semantically related context words of an NE). Therefore context vectors only include those content words whose POS tag weight is larger than 0.03 (corresponding to the top14 POS tags). We call them context vector (CV) words, and only consider these CV words in the location weight estimation.

Similar to the POS tag weights, location weights represent the semantic significance of CV words at different positions. Starting from a 20 word long window ranging from -10 (left 10 CV words) to 10 (right 10 CV

words), the weight corresponding to location l can be similarly estimated from the NE-word correlation coefficients:

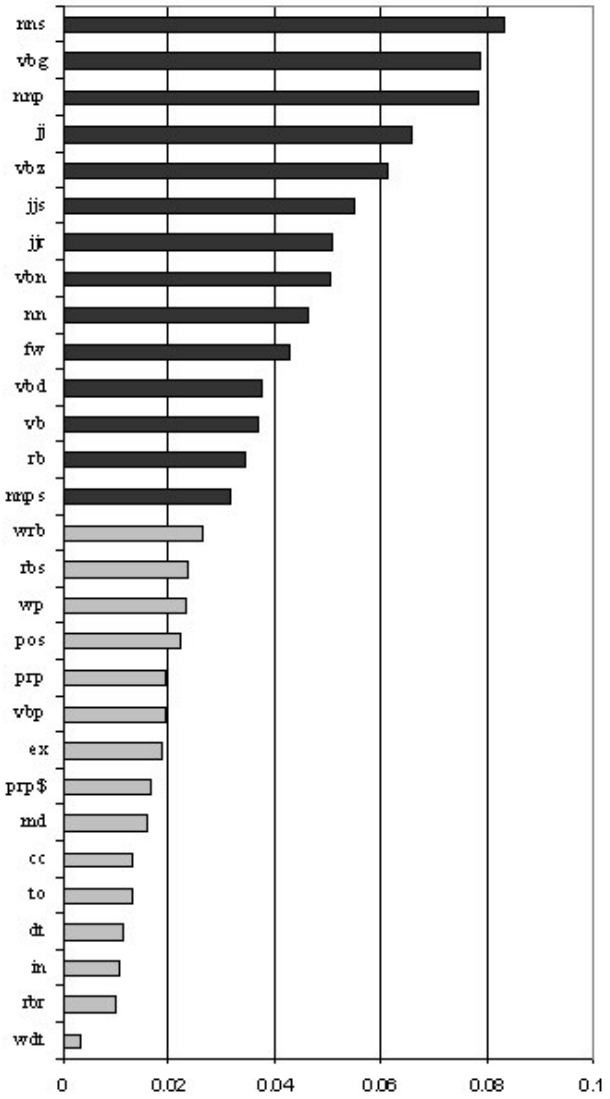

Figure 1. Normalized Word POS Weights

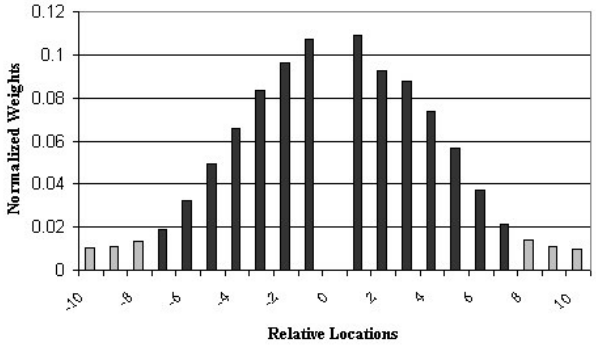

Figure 2. Normalized Word Location Weights

$$W(l) = \frac{1}{\sum_{(n,w)} C(n,w,l)} \sum_{(n,w)} C(n,w,l)\, p(w \mid n), \qquad (6)$$

where l is the location index, $l \in [-10,10], l \neq 0$. $C(n,w,l)$ is the frequency that word w occurs at the location l in the context vector of n.

Figure 2 illustrates the distribution of normalized location weights, which looks Gaussian: the closer a location is to the NE, the higher correlations it has. Notice that about 95% of weights are distributed within the [-7,7] window, so only the content words within this window are included in the context vector.

To summarize, the context vector of an NE is constructed from its left and right 7 content words, where "content words" are those whose POS tags are in the top 14 Content POS tag Set (CPS). The context vector is composed of both word identities and their semantic significance to the NE:

$$V = \{(w, W(t,l)) \mid l \in [-7,7], l \neq 0, t \in CPS\}, \qquad (7)$$

where $W(t,l) = W(t)W(l)$ is the product of their POS and relative location weights.

3.2 Semantic Similarity between Context Vectors

Given a source and target NE pair (n_e, n_f) with their context vectors (v_e, v_f), the semantic similarity between the two vectors can be defined as the "mutual translation probability", which is the product of two conditional semantic translation probabilities,

$$S(v_e, v_f) = P(v_f \mid v_e) P(v_e \mid v_f) \qquad (8)$$

where $P(v_e \mid v_f)$ is regarded as the probability that the source vector is "semantically translated" into the target vector. It is computed with a modified IBM translation model-2 [Brown et al. 1993],

$$P(v_e \mid v_f) = \frac{1}{I^J} \prod_{j=1}^{J} [W(t_j, l_j) \sum_{i=1}^{I} p(e_j \mid f_i)] \qquad (9)$$

where I is the length of the source vector and J is the length of the target vector. With this formula, each word in the source vector can be translated from any word in the target vector. The word translation probability is also adjusted by the POS and location weights of the target word, which emphasize the correct translations of important context words, for example the title of a person. $p(e \mid f)$ is the word translation probability estimated from a Chinese-English aligned corpus with IBM model 1. $P(v_f \mid v_e)$ is estimated in the similar way.

4 Cross-lingual Retrieval for NE Translations

Two similarity measures have been introduced to find NE translation pairs: pronunciation similarity based on a surface string transliteration model and semantic similarities based on a context vector semantic model. In this section, we will demonstrate how to apply these measures to search for NE translation pairs using the cross-lingual retrieval approach.

Given a Chinese NE together with the context in which it occurs (e.g., a document), we want to find English documents containing the NE translation, such that after automatically tagging all NEs in the retrieved text, we can compare the source NE with each English NE based on their phonetic and semantic measures, and ultimately choose the best-matched English NE as the translation. Assuming that documents containing the same NE share common topics (even if the texts are from different languages), our task is to search for topic-relevant English documents using the Chinese document as the query.

4.1 Query Generation

Given the source document, the query for a target NE translation search can be flexible: a few key phrases around the NE, the sentence holding the NE, or even the whole document. Containing less irrelevant information, short queries usually can generate less unrelated target text. However the identification and translation of key source phrases are crucial: if the query is not carefully selected or correctly translated, retrieved documents may not contain the target NE translation. On the other hand, long queries such as a sentence or the whole document may be less focused but with richer context, and the danger of missing relevant documents and correct NE translations is also reduced.

Due to the high risk of missing correct NE translation because of errors in identifying and translating source key phrases, we prefer to choose a longer context as the query, such as the whole document. In our current implementation, we use a statistical machine translation system to translate the Chinese document into English, after that feed the translation hypothesis into any search engine, such as Google or the Lemur Toolkit.

4.2 Corpus Indexing and Search Engine

Most commercial search engines have the advantage of accessing a large corpus and collecting huge information from web pages on the World Wide Web, which is very helpful for rare NE translations. However for our research purposes at this stage we prefer a more flexible corpus indexing strategy allowing both sentence-based and document-based indexing. So we start by building our own search engine using Lemur (Ogilvie and Callan 2002), a toolkit for language modeling and information retrieval.

The indexed corpus is composed of 963,478 English documents from the Xinhua News Agency, which corresponds to over 7.3 million sentences and 200 million words. The indexing just follows the standard procedure where no stemming and stop word removal is used. The retrieval model is the widely used TF-IDF model.

Given a query, the search engine returns a ranked list of relevant sentences or documents with relevance scores. We experiment with both sentence-based and document-based query generation and corpus indexing. From a test data of Chinese newswire documents, we selected 114 Chinese NEs and manually translated them, then we used our MT system to translate the Chinese sentences/documents containing these NEs into English. Considering that rarely occurring NEs are the most difficult to translate, the translated NEs are mostly incorrect in the translation hypotheses. These English hypotheses are fed into the search engine as the queries, and the top 1000 English sentences or documents are selected as the relevant text. We evaluated NE coverage by counting how many correct NE translations can be found in the retrieved texts, and it turned out that the document-based query/indexing covered about 70% of correct NE translations, while the sentence-based query/indexing has the coverage of about 60%. The reason may be that the topic information provided by each sentence is rather limited, and if its translation hypothesis is not reliable, the generated query could be severely distorted from the original meaning, thus the retrieved text may become irrelevant. In the following experiments we only use document-based querying and indexing.

4.3 Combining Similarity Features for NE Translation Selection

English NEs in the retrieved text are automatically tagged using IdentiFinder™, the NE tagging tool from BBN (Bikel et al., 1997). For each tagged English NE, its context vector is created according to its neighbor content words, with their POS tags and locations, as described in section 3.1.

To find the translation of a source NE n_f, we compare it with each tagged NE in the retrieved English text, using both transliteration similarity and context vector semantic similarity. We create the context vector for both the source NE and each tagged target NE. For each source and target NE pair (n_f, n_e), with their corresponding context vectors (v_f, v_e), their overall similarity score is defined as:

$$D(n_f, n_e) = \lambda_t P_{ne}(n_e \mid n_f) + \lambda_s S(v_f, v_e), \qquad (10)$$

where P_{ne} is the transliteration probability as computed in formula (3) and S is the context vector semantic similarity as computed in formula (8), λ_t and λ_s are the weights of each model empirically chosen based on experiment. The NE pairs with the highest overall similarity scores are considered translations. In practice, one source NE can be translated in several different ways, which have similar pronunciations but different spellings, and some of them are just typos. To make sure that translated NEs follow most people's usage, from among the top NE hypotheses with similar spelling, we choose the one with the highest frequency as the translation.

Figure 3 illustrates the overall architecture of the NE translation. In addition to various modules and data flows described above, one may notice the link from the NE Translation Selector to the Machine Translation module, which indicates that translated NE pairs can be further integrated into the machine translation engine to improve the query translation quality, retrieve better relevant documents and improve NE translation again, and this can be an iterative process.

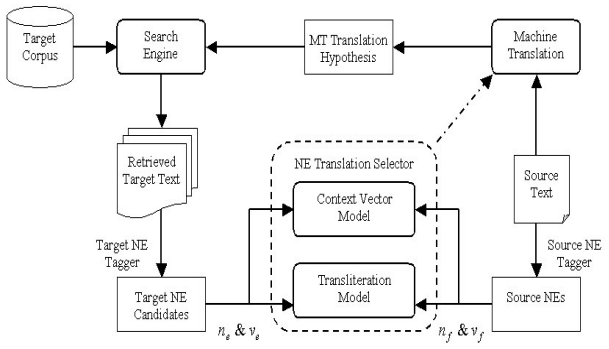

Figure 3. Overall Architecture of NE Translation

5 Experiment Results and Discussion

5.1 Transliteration Model Training and Evaluation

We train the Chinese-English surface string transliteration model using the manually compiled Chinese-English dictionary provided by LDC, which contains English translations for 54,131 unique Chinese words. Initially, 3,000 word translation pairs with a minimum string matching cost are extracted, under the 0/1 cost function. Most of them are NE pairs whose pinyin format resembles the English translation. The initial letter transliteration model and Chinese character language model are trained from this name list. Using these models, an additional 500 NE pairs with a minimum transliteration cost are extracted in each iteration, and added into the existing NE pair list to update models. This process continues until adding more NE pairs does not improve the extraction accuracy further, which usually happens at the 5-6th iteration where a total of 5,500~6,000 NE entries are included.

Iter.	0/1	1	2	3	4	5	6
Prec.	93.1	96.0	95.8	97.8	99.1	99.1	99.1

Table 1. Precision of Top 6000 NE Pairs Using Different Models

Table 1 presents the NE extraction precision of the top 6000 NE pairs using a different model in each iteration. "0/1" represents the result when using only the starting 0/1 cost function. One can notice a trend of increasing precision after each iteration, although the increase is smaller and smaller until negligible at the 5~6th iteration, indicating that most NE pairs in the dictionary have already been included and that adding more non-NE entries will not benefit the transliteration model.

5.2 Creating Context Vectors

In section 3.1, the NE-word correlation coefficients are estimated from a subset of the indexed English Xinhua News corpus. It is composed of over 37 million words from 188,755 documents. 380,641 unique English NEs are automatically tagged, and the coefficient is calculated for each (NE, word) pair.

prime	0.0319	clinton	0.0046
israeli	0.0222	bill	0.0034
minister	0.0194	yatom	0.0032
caretaker	0.0160	david	0.0030
yasser	0.0145	summit	0.0030
arafat	0.0115	ariel	0.0029
leader	0.0069	camp	0.0028
palestinian	0.0063	likud	0.0028
outgoing	0.0060	sharon	0.0028
al-shara	0.0047	cabinet	0.0027

Figure 4. Context Words with High Correlation Coefficients for the NE "Ehud Barak"

Figure 4 shows the top 20 words/coefficients for the NE "Ehud Barak", the former Israeli Prime Minister. It shows that words with high coefficients are mostly topic relevant words, which indicates that the Phi-square based NE-word correlation coefficient is an effective measure of topical relevance.

In the following example, we will show a Chinese NE (伊 辛)'s context vector created from a Chinese sentence and the best-matched English NE (**Otmar Issing**)'s context vector created from the retrieved text,

then illustrate the semantic correlations between the two vectors.

Chn: 欧洲/NR 中央银行/NN 首席/NN 经济学家/NN 伊/NR 辛/NR 博士/NN 20/CD 日/NR 在/P 此间/NN 表示/VV…

Eng: European/JJ Central/NNP Bank/NNP Chief/NNP Economist/NNP **Otmar**/NNP **Issing**/VBG told/VBD the/DT European/JJ Parliament/NNP last/JJ week/NN that/IN …

In the above sentences, NEs are automatically tagged and highlighted for each language. The POS information has been automatically tagged as well, where the taggers are trained from some manually annotated data for each language using transformation-based learning (Brill 1995). Considering POS and distance to the NE, the context vectors (words and their normalized weights) for the Chinese NE (left) and English NE (right) are shown in Figure 5. The links between Chinese and English words indicate they are translations of each other. In this example, links between words with high semantic weights show a strong correlation between the two context vectors.

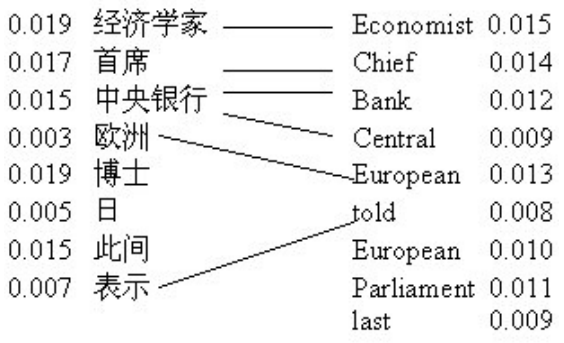

Figure 5. Context Vector of Chinese NE (*left*) and Best-matched English NE (*right*)

5.3 Improving Machine Translation Quality with NE Translation

To evaluate the effectiveness of the proposed NE translation strategy, we test it for a Chinese-English machine translation task. The test dataset is the NIST 2002 Machine Translation Evaluation test data. The test data is composed of 100 Chinese documents, 878 sentences, and 25,430 words. 2469 NEs are automatically tagged, and among them PERSON, LOCATION and ORGANIZATION names roughly account for 20%, 60% and 20% respectively. Since most ORGANIZATION NEs are semantically translated word-by-word, and since we already have good word and phrase translation components in the baseline system, we will focus on PERSON and LOCATION NE translations, as they are often transliterated.

The baseline system incorporates several word and phrase transducers for text translation: a 50K entry word-based C-E translation lexicon from LDC, which has the best word translation accuracy because of manual verification; several phrase transducers automatically constructed from a 6M words bilingual corpus using HMMs and integrated segmentation and alignment approaches (Vogel et. al. 2003). Importantly, the baseline also includes a 39K entry NE transducer which is constructed by aligning tagged NEs from the same parallel corpus according to multiple NE alignment costs (Huang et. al. 2003).

Among 1,898 tagged PERSON and LOCATION NEs, 400 NEs are not covered by the LDC translation lexicon. After manually removing incorrectly tagged NEs, 338 true NEs (corresponding to 158 unique NEs) are translated with the transliteration model plus the semantic context vector model, and the translation hypotheses are compared with the reference translations for evaluation.

Table 2 shows the type and token NE translation precision using different similarity models, where "**Translit**" means using the transliteration model only, and "+**SCV**" means additionally combining the context vector semantic model. It also shows the performance of the baseline system, where the translations basically come from several phrase and NE transducers trained from the 6M words bilingual corpus. The limited parallel corpus coverage explains the relatively lower performance of the Baseline system, as the source NEs cannot be found in the parallel corpus. When finding NE translations from the retrieved monolingual text, the surface string transliteration model alone increases the translation precision by about 30%, and the context vector semantic model additionally improves the translation accuracy by about 10%. Further error analysis indicates that 50% of errors are due to the limited coverage of retrieved documents, i.e., correct NE translations are either not included in or not retrieved from the indexed English corpus.

	Token (338) Precision	Type (158) Precision
Baseline	27.8%	27.8%
+Translit	57.1%	50.0%
+SCV	67.8%	59.5%

Table 2. NE Translation Precision

We integrate both sets of NE translation hypotheses into the baseline system: "+**Translit**" and "+**SCV**", and test them in different translation tasks: the small data track and the large data track differing in the amount of bilingual resources allowed for use. To accurately measure the contribution of the proposed NE translation method, we first extract 164 sentences containing these rarely occurring NEs from the whole test set (887 Chinese sentences), translate and evaluate on this subset,

then we evaluate the NE translations on the whole test data. The translation quality is measured by the automatic MT evaluation metrics, such as NIST and Bleu scores.

Table 3 shows the translation scores of different system configurations on the NE sentences subset, and table 4 shows the translation scores on the whole test data. Because the selected sentences are hard to translate due to these rarely occurring NEs, their translations have lower NIST and Bleu scores than the whole test set (1.0 difference in NIST and 0.03 difference in Bleu for the Baseline). When adding transliterated NE translations, an obvious improvement can be observed in all the cases. Additionally adding the context vector model also leads to a small but consistent improvement.

	Small track		Large track	
	NIST	Bleu	NIST	Bleu
Baseline	5.6234	0.1166	6.8483	0.1794
+Translit	6.2684	0.1387	7.1969	0.2005
+SCV	6.3618	0.1404	7.2779	0.2025

Table 3. C-E MT Evaluation on NE Sentences Subset

	Small track		Large track	
	NIST	Bleu	NIST	Bleu
Baseline	6.5765	0.1479	7.8733	0.2023
+Translit	6.7718	0.1537	7.9573	0.2075
+SCV	6.8702	0.1580	7.9790	0.2079

Table 4. C-E MT Evaluation on Whole Test Set

6 Conclusion

We propose an approach to translate rarely occurring NEs by combining their phonetic and semantic similarities . Given a source NE and its context, this approach generates queries in the target language according to the context translation hypotheses, then searches for relevant documents from a target corpus. Target NEs in retrieved documents are compared with the source NE based on their phonetic and contextual semantic similarities, and the best-matched one is selected as the correct translation. Experiments show that this approach achieves 67% on translation accuracy, and consistently improves the translation quality on different tasks.

References

Y. Al-Onaizan and K. Knight. Translating named entities using monolingual and bilingual resources. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* pp400-408, Philadelphia, PA, July, 2002.

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.

E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4): 543--565. 1995.

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. 1993.

N. A. Chinchor. Overview of MUC-7/MET-2. *In Proceedings of the Seventh Message Understanding Conference(MUC-7),* Fairfax, VA, April, 1998.

F. Huang, S. Vogel and A. Waibel. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. *In Proceedings of the ACL'03, Workshop on Multilingual and Mixed Language Named Entity Recognition*. Sapporo, Japan, July, 2003.

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

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR* 163(4) p845-848, 1965.

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.

R. C. Moore. Learning translations of named-entity phrases from parallel corpora. *In Proceedings of 10th Conference of the European Chapter of ACL*, Budapest, Hungary, 2003.

P. Ogilvie and J. Callan. Experiments using the Lemur toolkit. *In Proceedings of the 2001 Text REtrieval Conference (TREC 2001)*. pp. 103-108, 2002. http://www.cs.cmu.edu/~lemur

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*. Montreal, Quebec, Canada, 1998.

S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venogupal, B. Zhao and A. Waibel. The CMU statistical machine translation system. *In Proceedings of the MT Summit IX Conference* New Orlean, LA, September, 2003.