Named Entity Translation with Web Mining and Transliteration

Long Jiang*, Ming Zhou*, Lee-Feng Chien[+], Cheng Niu*
*Microsoft Research Asia
*5F Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing 100080, PRC
[+]Institute of Information Science, Academia Sinica, Taiwan
*{longj, mingzhou, chengniu}@microsoft.com, [+]lfchien@iis.sinica.edu.tw

Abstract

This paper presents a novel approach to improve the named entity translation by combining a transliteration approach with web mining, using web information as a source to complement transliteration, and using transliteration information to guide and enhance web mining. A Maximum Entropy model is employed to rank translation candidates by combining pronunciation similarity and bilingual contextual co-occurrence. Experimental results show that our approach effectively improves the precision and recall of the named entity translation by a large margin.

1 Introduction

Named entity (NE) translation accepts a named entity from a source language as input and outputs its translations into a target language. For instance, *"Pennant"->"彭南特"* (English to Chinese). Large quantities of new named entities appear each day in newspapers, web sites and technical literature, but their translations normally cannot be found in translation dictionaries. Improving named entity translation is important to translation systems and cross-language information retrieval applications. Moreover, it benefits bilingual resources acquisition from the web and translation knowledge acquisition from corpora. While named entities normally refer to a range of concepts like people names, place names, organization names, product names etc., the focus of this paper is the translation of people names from English to Chinese.

Many of the previous works extract named entity translation dictionaries from bilingual corpora [Kupiec, 1993; Feng et al., 2004]. To alleviate the strong dependency on parallel corpora, [Rapp, 1999; Fung and Yee, 1998] try to obtain named entity translations from nonparallel, comparable texts, or unrelated bilingual corpora.

Since a large proportion of English named entities can be translated by transliteration, many works try to build transliteration models with a rule-based approach [Wan and Verspoor, 1998] and statistics-based approach [Knight and Graehl, 1998; Lin and Chen, 2002; Virga and Khudanpur, 2003; Gao, 2004]. Both approaches, however, still have room for improvement. Rule-based approaches adopt linguistic rules for a deterministic generation of translation. However, it is hard to systematically select the best Chinese character from multiple Chinese characters that have the same pronunciation, such as *"爱"* and *"埃"*. Statistics-based transliteration approaches select the most probable translations based on knowledge learned from the training data. This approach, however, still may not work perfectly when there are multiple standards [Gao, 2004]. For example, "ford" at the end of an English NE is transliterated into *"福德"* in most cases (e.g., *"Blanford"-> "布兰福德"*), but sometimes, it is transliterated into *"福"* (e.g., *"Stanford"->"斯坦福"*). Obviously, there is significant flexibility in transliteration generation of foreign names in real world, and transliteration selection is somewhat subjective. Hence, translation relying on the statistical machine transliteration only may not work well.

The web contains an enormous dataset of languages, many of which are instantly available and up to date. In recent years, some researchers, like [Wang et al., 2004; Cheng et al., 2004; Nagata et al., 2001; Zhang et al, 2005] have used it to extract translations with promising results. In particular, [Wang et al., 2004; Cheng et al, 2004] propose a novel approach. They search target language web pages using the source term or NE. Then it extracts the translation candidates based on *SCPCD*[1] score + Local Maxima algorithm and ranks the generated candidates with Chi-Square method and context vector method. This approach gets excellent results for high-frequency terms and NEs. However, it cannot handle the low-frequency words or ambiguous words (e.g. *"Younger", "Red"*) well, because their translations scarcely appear in the search results.

Other efforts have been made to undertake combinations of transliteration and corpora mining [Shao and Ng, 2004; Huang et al., 2004] or web mining [Al-Onaizan and Knight, 2002]. In particular, [Al-Onaizan and Knight, 2002] use

[1] $$SCPCD(w1...wn) = \frac{LC(w1...wn)RC(w1...wn)}{\frac{1}{n-1}\sum_{i=1}^{n-1} freq(w1...wi)\,freq(wi+1...wn)}$$

Where *w1…wn* is the n-gram to be estimated, *LC(w1…wn)* is the number of unique left adjacent characters. *RC(w1…wn)* is the number of unique right adjacent characters. *freq(wi…wn)* is the frequency of the N-gram *w1…wn*. [Cheng et al, 2004].

statistical machine transliteration model to generate translation candidates, and use the web for translation identification. This method demonstrates positive results when using the web for selecting correct translation from candidates, but not for helping generate translation candidates.

In this paper, we are interested in the question of : how to and to what extent the NE translation can benefit from the combination of transliteration and web mining in both candidate generation and translation identification. Specifically, we are concerned with how to enhance the transliteration results with web information. That means, for example, *"科林顿"* (output of transliteration system for *"Clinton"*) could be corrected as *"克林顿"* (the correct translation of *"Clinton*"), and how to guide web mining with transliteration information, rather than using statistical association alone. To do this, we propose a novel framework of NE translation that effectively incorporates approaches of transliteration and web mining. Specifically, we build a transliteration model first, and the transliteration result is used to guide web mining for generating translation candidates. Furthermore, a Maximum Entropy (ME) model is designed to rank the translation candidates with various relevant features.

Benchmarks are performed on each component and the overall system to reveal our improvement to previous works. Experimental results show that NE translation can be consistently improved with our new approach by a large margin.

1. Using transliteration-guided web mining, the translation candidate coverage is improved from 54.5% to 74.5%.
2. By including pronunciation similarity and bilingual contextual co-occurrence features, a ME model help to improve the overall translation precision from 18.5% to 47.5% in top 1 answer.

The remainder of this paper is structured as follows. Section 2 presents steps of our approach. In Section 3, the transliteration model is described. Section 4 introduces our new approach to provide translation candidates by combining transliteration with web mining. The ranking method for translation candidates with a ME model is presented in Section 5. Experiments and performance comparison with previous works are reported in Section 6. The last section points out the directions of future work and concludes our study.

2 Our Approach

Specifically, the translation process we proposed can be described as follows:

1. Transliteration: To transliterate English NE into Chinese word base on their pronunciation similarity. In this step, a three-level transliteration model is built: English surface string to Chinese Pinyin string, Chinese Pinyin string to Chinese character string and Chinese character language model.
2. Web mining for translation candidates enhanced by transliteration: Compared with [Wang et al., 2004; Cheng et al, 2004], transliteration-based similarity is used to improve the translation candidate generation for low-frequency words. The Chinese pages are searched using the input English NE as query. The n-gram strings are selected from the snippets as translation candidates if they are pronounced similarly to the input NE.
3. Web mining for translation candidates using queries expanded by transliteration: The input English NE is combined with each Chinese character (anchor) in the transliteration results to form expanded queries. Then the expanded queries are used to search web pages. From the returned snippets, we select the n-grams which contain the anchors and have strong pronunciation similarity to the input NE as additional translation candidates. This approach aims to deal with ambiguous words whose majority usage is not a proper name (e.g. *"Younger"*).
4. Finally, all the translation candidates generated from the step 2 and step 3 are ranked by a ME model that has features covering pronunciation similarity and bilingual contextual co-occurrences.

3 Transliteration

Given an English NE, denoted as *E*, we first syllabicate it into a "syllable" sequence *SE = {e1, e2 ... en}* with the following linguistic rules we defined:

1. *a, e, i, o, u* are defined as vowels. *y* is defined as a vowel when it is not followed by a vowel. All other characters are defined as consonants;
2. Duplicate the nasals *m* and *n* whenever they are surrounded by vowels. And when they appear behind a vowel, they will be combined with that vowel to form a new vowel;
3. Consecutive consonants are separated;
4. Consecutive vowels are treated as a single vowel;
5. A consonant and a following vowel are treated as a syllable;
6. Each isolated vowel or consonant is regarded as an individual syllable.

For example, *"Campanelli"* is split into *"cam / pan / ne / l / li"*. *"Clinton"* is split into *"C / lin / ton"*. *"Lasky"* is split into *"La / s / ky"*. *"Meyerson"* is split into *"Me / ye / r / son"*.

Then a generative model is used to transliterate the syllabicated English name into Chinese character string, which follows [Knight and Graehl, 1998]. Specifically, for the generated "syllable" sequence *SE = {e1, e2 ... en}*, we seek a Chinese Character sequence *C = {c1, c2 ... cm}* that maximizes the product of *p(SE|PC)*, *p(PC|C)* and *p(C)*, *i.e.,*

$$C^* = \arg\max_c p(SE|PC)p(PC|C)p(C)$$

where *PC* is a Chinese Pinyin sequence, *p(SE|PC)* is the probability of translating *PC* into *SE*, *p(PC|C)* is the probability of translating *C* into *PC* and *p(C)* is the generative probability of a character-based Chinese language model.

Different from [Knight and Graehl, 1998; Virga and Khudanpur, 2003] whose transliteration model is based on phoneme, we use "syllable" as translation units (TU).

Here, a Chinese "syllable" means a Chinese Pinyin string that is corresponding to a Chinese character. And English "syllable" means a combination of several English letters that can generate a corresponding Chinese Pinyin "syllable".

Using syllables as TU has the following advantages: 1) compared with phoneme-based approach, a syllable has far less ambiguity in finding the corresponding Chinese Pinyin string. For instance, *"kem"* has fewer corresponding Chinese Pinyin strings than the phoneme sequence */k eh m/*; 2) a syllable always corresponds to a legal Pinyin sequence.

The translation model *p(SE|PC)* can be trained with GIZA++[2] on parallel English names and the Pinyin representation of Chinese names. The translation model *p(PC|C)* can be approximately obtained from a pronunciation dictionary. In our experiment, we used an in-house dictionary. The language models *P(C)* is approximated by using a trigram model and trained with a list of Chinese transliterated names.

4 Web Mining

After we get the transliteration result of source NE, we want to use it to help finding the translation candidates on the web. The specific methods are as follows.

4.1 WM-NE: Candidate Collection by Searching the Web with NE

Same with [Wang et al., 2004], we use the input English NE as a query to search for the Chinese pages and extract candidates in the returned top 100 snippets. In our experiments, Google search engine[3] is used for obtaining the snippets.

It is noticed that usually the results of statistical transliteration model are not exactly correct, but very similar with the correct translation. If we use web data to correct the transliteration result, the accuracy would be improved a lot. Based on this insight, the n-grams that have similar pronunciation with the transliteration results are collected from the search engine snippets, and are used for transliteration correction.

Suppose *W* is the best transliteration result output by previous transliteration model. We define the score of an n-gram *wn* being a translation candidate as the product of *TL(wn, W)* and *p(wn)*, *i.e.*,

$$\mathrm{Score}(\mathrm{wn}) = \mathrm{TL}(\mathrm{wn}, \mathrm{W}) \times \mathrm{p}(\mathrm{wn})$$

where, *TL(wn, W)* indicates the pronunciation similarity between the n-gram *wn* and the source English NE, and *p(wn)* indicates the probability of *wn* being a transliterated name which is the generative probability of the Chinese character-based language model used in Section 3. And *TL(wn, W)* can be computed using the following formula.

$$TL(wn, W) = 1 - \frac{ED(PYwn, PYw)}{L(PYw) + L(PYwn)}$$

where *PYw* is the Pinyin sequence of *W* and *PYwn* is the Pinyin sequence of *wn*. *L(PYwn)* and *L(PYw)* are the lengths of *PYwn* and *PYw* respectively, and *ED(PYwn, PYw)* is the Edit Distance between *PYw* and *PYwn*.

Based on the formulas above, we compute the scores for all n-grams in the snippets. And then select those n-grams as candidate whose score of being a translation candidate is over a threshold.

This approach can make adequate use of the transliteration information obtained in the previous model and hand those transliterated translation even though it is a low-frequency word in the snippets. By the way this method is also independent of NE tagger.

4.2 WM-NE-Anchor: Enhancing Web Mining Using Transliteration Result as Anchors

In addition to WM-NE, the result of transliteration $C^* = \{c_1, c_2 \ldots c_m\}$ is used as anchor information to guide the web mining to obtain additional translation candidates by using the following steps.

1. Search the web with an anchor character and the input NE. Each character, c_i, is combined with source English NE as a query to search for web pages. The first 30 snippets are selected;
2. Surrounding the position of c_i in a snippet, we extract all the n-gram character strings that contain c_i and its score of being a translation candidate described in Subsection 4.1 is over a threshold as candidates.

For example, *"Nikos"* is transliterated into *"尼克斯"* by our transliteration model in Section 3. Then the Chinese word is split into three characters: *"尼"*, *"克"* and *"斯"*. Each of these characters is combined with *"Nikos"* to form a Boolean query to search for web pages. For each query, we select the top 30 returned snippets to form a small corpus, e.g. for the query *"尼" + "Nikos"*, we get a corpus { *副部长 尼科斯·法尔马基斯Deputy Minister: Nikos Farmakis* }. In the corpus we search the position where *"尼"* appears, and select all n-gram strings that contains *"尼"* and has score higher than the threshold to be a translation candidate, and the results are { *尼科, 尼科斯* }.

This approach can effectively enhance WM-NE. In WM-NE, the returned snippets may not contain the anchor + NE cases within 100 snippets. One of the reasons is that the source NE may be quite ambiguous. It can be used as a person name sometimes, but this is not its majority usage. For example, "Spain" can be translated into "*斯佩恩*" when used as a person name, but more popularly, it should be translated into "*西班牙*". When we search web using only "Spain" as query, all returned snippets by Google do not contain "*斯佩恩*", but when we search web using "*Spain + 斯*", the top 30 returned snippets contain "*斯佩恩*" ({ *道格拉斯·斯佩恩·Douglas Spain* }). So the WM-NE-Anchor method can find the specific translation of many ambiguous words.

5 Ranking Translation Candidates

A ME model is used to rank the translation candidates obtained above, which contains the following four features

[2] http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html.

[3] http://www.google.com

functions *hi(C, E) (i = 1, 2, 3, 4)*. Where, *C* denotes a Chinese candidate, *E* denotes the source English NE.

1. Same with [Cheng et al., 2004], the Chi-Square of translation candidate *C* and the input English named entity *E* is used as one of our features. The Chi-Square is computed using the formula below:

$$h1(C,E)=\frac{N\times(a\times d-b\times c)^2}{(a+b)\times(a+c)\times(b+d)\times(c+d)}$$

- a = the number of pages containing both *C* and *E*
- b = the number of pages containing *C* but not *E*
- c = the number of pages containing *E* but not *C*
- d = the number of pages containing neither *C* nor *E*
- N = the total number of pages, *i.e.*, $N = a+b+c+d$

In our experiments, *N* is set to 4 billion. Actually, the value of *N* doesn't affect the ranking once it's positive. To get the value of parameters in the formula above, we combine *C* and *E* as a query to search with Google for web pages. And the returned page contains the number of total pages containing both *C* and *E* which is corresponding to a in the formula above. Then we use *C* and *E* as queries respectively to search the web and we can get the two numbers *Nc* and *Ne*, which represent the number of pages contain *C* and *E* respectively. Then we can compute $b= Nc - a$, $c= Ne - a$ and $d= N - a - b - c$.

2. Contextual feature *h2(C, E)=n*, where *n* is the number of occurrence, in any of the snippets selected, that *E* is in a bracket following *C* or *C* is in a bracket following *E*;
3. Contextual feature *h3(C ,E)=n* where *n* is the number of occurrence, in any of the snippets selected, *C* and *E* are immediate neighbors;
4. Similarity of *C* and *E* in terms of transliteration similarity *h4(C, E)*, which is equal to *TL(C, E)* described in Subsection 4.1.

With these features, the ME model is expressed as:

$$P(C\mid E)=p_{\lambda_1^4}(C\mid E)=\frac{\exp[\sum_{m=1}^{4}\lambda_m h_m(C,E)]}{\sum_{C'}\exp[\sum_{m=1}^{4}\lambda_m h_m(C',E)]}$$

To estimate the parameters λ_i we use the Generalized Iterative Scaling algorithm of [Darroch and Ratcliff, 1972].

6 Experiments

We use the same method as [Gao, 2004] to collect the training data, *i.e.*, we extract all named entity pairs from the LDC Chinese-English Name Entity Lists Version 1.0[4], which also appear in CMU pronunciation dictionary[5]. We finally obtain 25,718 person names. Out of these name pairs, 200 are selected as development set, 200 as testing set, and the rest is for training.

[4] Catalog Number by LDC: LDC2003E01

[5] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

6.1 Transliteration Evaluation

To evaluate our transliteration model using syllables as translation units, we compare its performance with [Virga and Khudanpur, 2003] which is also on English to Chinese transliteration. The measures are Pinyin error rate (Pinyin ER) and Chinese character error rate (Char ER). Here, Pinyin ER is the Edit Distance between the "correct" Pinyin representation of the correct transliteration and the Pinyin sequence output by the system [Virga and Khudanpur, 2003]. Char ER is defined similarly, *i.e.*, it is the Edit Distance between the character sequence of the "correct" transliteration and the character sequence output by the system. The results are listed below:

	Pinyin ER	Char ER
[Virga and Khudanpur, 2003]	42.5%	N/A
Our transliteration model	39.6%	46.1%

Table 1 Transliteration model evaluation

From Table 1, we can see that our model outperforms [Virga and Khudanpur, 2003]. The results support using "syllable" as translation unit in both sides in transliteration model. In our experiments, the TU on English side contains about 1,260 English syllables and the TU on Chinese side contains 370 Pinyin syllables.

6.2 Effectiveness of Candidate Generation by Combining Transliteration with Web Mining

We want to know whether our new web mining approach, *i.e.*, WM-NE and WM-NE-Anchor, improves the coverage of translation candidates. So we made some experiments to compare our method with [Al-Onaizan and Knight, 2002] and [Wang et al., 2004]. Because they used non-public data set for evaluation and [Al-Onaizan and Knight, 2002] performed Arabic-English translation instead of English-Chinese translation, which make the direct comparison difficult. Instead, we reimplemented their methods and benchmarked our improvement using LDC data set. The results are listed below:

	Candidates coverage	Average number of candidates
[Al-Onaizan and Knight, 2002]	46.5%	100
[Wang *et al.*, 2004]	54.5%	295.1
WM-NE	58%	64.0
WM-NE + WM-NE-Anchor	74.5%	144.0

Table 2 Candidates coverage comparison with previous approaches

In [Al-Onaizan and Knight, 2002], N-best outputs of transliteration model are used as translation candidates for re-ranking. Obviously, when *N* increases, the coverage of translation candidates increases. So which number should we assign to *N* to make a fair comparison? In our reimplemen-

tation experiment of their algorithm, we found that after N increases over 100, the increment of candidate coverage becomes negligible (see Figure 1). So we finally decide to set N equal to 100 for our comparison experiment.

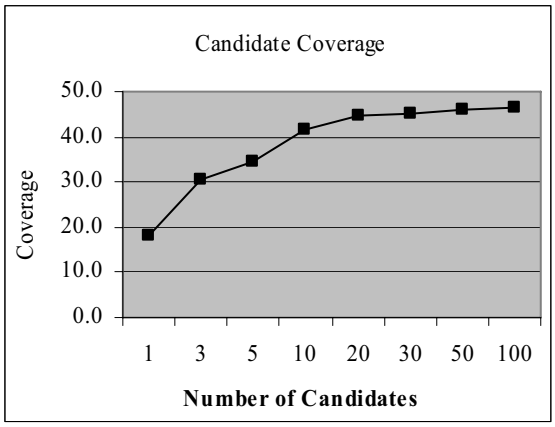

Figure 1 Candidates coverage of N best transliteration output

In [Wang et al., 2004], candidates are extracted by *SCPCD* + Local Maxima Algorithm. This method is good at handling high-frequency words, but worse for low-frequency words. In our testing data, the average frequency of the correct translation appearing in the search result is about eight, and for 40.5% of NEs, the frequency of its correct translation is less than three. So *SCPCD* + Local Maxima Algorithm is hard to extract these translations. From Table 2, we also notice that WM-NE-Anchor improved the coverage significantly. That means our WM-NE-Anchor method does work for many ambiguous NEs.

6.3 ME Model Evaluation

To test the effectiveness of each feature used in our ME model, we conduct four experiments. Like [Cheng et al., 2004], the metric is *top-n inclusion* which is defined as the percentage of the NE whose translations could be found in the first *n* extracted translations.

	Chi-Square	TL	Chi-Square + TL	All features
Top-1	7.5%	34.0%	43.0%	47.5%
Top-3	23.0%	55.5%	59.0%	61.5%
Top-5	33.0%	63.5%	63.5%	66.0%
Top-10	49.0%	69.5%	66.5%	69.5%

Table 3 Candidate ranking experiment result

Table 3 shows that incrementally adding features is effective in improving the *top-n inclusion*. Ranking with Chi-Square method alone, the precision is 7.5%. Ranking with TL score alone, the precision is 34.0%. Combining these two features together, precision reaches up to 43.0%, and using all features mentioned in Section 5, it reaches a 47.5% precision rate.

We further analyze the reasons. Chi-Square, as the statistical association of the English NE and the translation candidates, is effective mainly on high-frequency candidates, but less effective on low-frequency candidates. Moreover, Chi-Square cannot differentiate candidates based on contents. However, transliteration scores bring information on linguistic association between NE and candidates. Therefore, they can improve ranking effects. In addition, from our results we can see that contextual features are also helpful in improving rankings with about 4.5% for the *top-1 inclusion*.

6.4 Comparing Previous Approaches

To compare with the previous approaches for transliteration and web mining, three experiments are conducted:

1. Exp_1 (Based on [Al-Onaizan and Knight, 2002]): uses transliteration model to generate candidates and then combine straight web counts of candidates to select correct translations.
2. Exp_2 (Based on [Wang et al., 2004]) provides translation candidates via *SCPCD* + Local Maxima Algorithm and ranks them by Chi-Square method and Context vector method.
3. Exp_3 (Our approach) gets translation candidates via WM-NE and WM-NE-Anchor and ranks them with all of the four features mentioned in Section 5.

Table 4 indicates our new approach surpasses the previous approaches of transliteration and web mining by a substantial margin.

	Exp_1	Exp_2	Exp_3
Top-1 inclusion	18.0%	18.5%	47.5%
Top-5 inclusion	35.5%	35.0%	66.0%
Top-10 inclusion	41.5%	39%	69.5%

Table 4 Comparing previous approaches

To analyze errors, we checked a sample of 50 NEs. 24 (48%) NEs provide a correct translation. In the remaining data, 10 (20%) NEs acquired translations different from the ones defined in LDC data though they are acceptable. Further study shows that most are actually more popular than that LDC data. For instance, the translation of "Dockery" in LDC data is "*多克里*". Our system outputs "*道克里*". We check with Google, "*道克里*" and "*Dockery*" co-occurs in 119 pages, but "*Dockery*" and "*多克里*" co-occurs in only three pages. This may indicate that "*道克里*" is also a popular translation.

We further analyze the error sources for the remaining 16 (32%) incorrect NE translations. For five NEs, the correct translations do not co-occur in any Chinese web page. For eight NEs, the correct translations are not ranked high enough in the candidate sets by TL score. For the other three NEs, the translation errors result from mistakenly identifying lexical boundaries in the candidate generation stage. It should be noted that, since our testing data is randomly selected from a large LDC data set, and many of them are scarcely used in real world, the performance of our system should not be compared directly to those work based on their specific testing data.

7 Conclusions

In this paper, we present a new approach to combine transliteration and web mining for NE translation. Experimental results show that our approach effectively improves the precision and recall of the named entity translation by a large margin.

Our approach has the following contributions:

1. Translation candidate generation is enhanced by combining transliteration-based similarity and web mining. Transliteration is especially helpful for low-frequency word translation.
2. Using queries expanded by transliteration, web mining can further improve the coverage of candidate generation.
3. A ME model incorporating different knowledge is effective in ranking translation candidates.

In the future, we want to extend this approach to NE translation for other language pairs. We are also interested in adapting the method to translate terminologies.

References

[Al-Onaizan and Knight, 2002] Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In Proc. of ACL-02. pp: 400-408.

[Cheng et al., 2004] Pu-Jen Cheng, Wen-Hsiang Lu, Jer-Wen Teng, and Lee-Feng Chien. 2004. Creating Multilingual Translation Lexicons with Regional Variations Using Web Corpora. In Proc. of ACL-04. pp. 534-541.

[Darroch and Ratliff, 1972] J.N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics. Vol 43. pp. 1470-1480

[Feng et al., 2004] Donghui Feng, Yajuan Lv, and Ming Zhou. 2004. A New Approach for English-Chinese Named Entity Alignment. In Proc. of EMNLP-2004. pp. 372-379

[Fung and Yee, 1998] Pascale Fung and Lo Yuen Yee. 1998. An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In Proc. of the 36th Annual Conference of the ACL, pp 414-420.

[Gao, 2004] Wei Gao. 2004. Phoneme-based Statistical Transliteration of Foreign Names for OOV Problem. A Thesis of Master. The Chinese University of Hong Kong.

[Huang et al., 2004] Fei Huang, Stephan Vogel and Alex Waibel. 2004. Improving Named Entity Translation Combining Phonetic and Semantic Similarities. HLT-NAACL 2004: 281-288.

[Knight and Graehl, 1998] Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. Computational Linguistics 24(4): 599-612.

[Kupiec, 1993] Julian Kupiec. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In Proc. of the 31st Annual Meeting of the ACL. pp. 17-22.

[Lin and Chen, 2002] Wei-Hao Lin and Hsin-Hsi Chen. 2002. Backward Machine Transliteration by Learning Phonetic Similarity. In Proc. of the 6th CoNLL, pp. 139-145.

[Nagata et al., 2001] Masaaki Nagata, Teruka Saito, and Kenji Suzuki. 2001. Using the Web as a Bilingual Dictionary. In Proc. of ACL 2001 Workshop on Data-driven Methods in Machine Translation, pp. 95-102.

[Rapp, 1999] Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In Proc. of ACL-99, pp. 519-526.

[Shao and Ng, 2004] Li Shao and Hwee Tou Ng. 2004. Mining new word translations from comparable corpora. In Proc. of Coling 2004, pp. 618–624

[Virga and Khudanpur, 2003] Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In Proc. of the ACL Workshop on Multi-lingual Named Entity Recognition. pp. 57-64.

[Wan and Verspoor, 1998] Stephen Wan and Cornelia Verspoor. 1998. Automatic English-Chinese Name Transliteration for Development of Multilingual Resources. In Proc. of COLING-ACL 1998, pp. 1352-1356.

[Wang et al., 2004] Jenq-Haur Wang, Jei-Wen Teng, Pu-Jen Cheng, Wen-Hsiang Lu, Lee-Feng Chien. 2004. Translating unknown cross-lingual queries in digital libraries using a web-based approach. In Proc. of JCDL 2004, pp. 108-116.

[Zhang et al., 2005] Ying Zhang, Fei Huang, Stephan Vogel. 2005. Mining translations of OOV terms from the web through cross-lingual query expansion. SIGIR 2005: 669-670.