A Statistical Approach to Chinese-to-English Back-Transliteration

Chun-Jen LEE[1,2] **Jason S. CHANG**[2] **Jyh-Shing Roger JANG**[2]

[1] Telecommunication Labs.
Chunghwa Telecom Co., Ltd.
Chungli, Taiwan, R.O.C.
cjlee@cht.com.tw

[2] Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
{jschang, jang}@cs.nthu.edu.tw

Abstract

This paper describes a statistical approach for modeling Chinese-to-English back-transliteration. Unlike previous approaches, the model does not involve the use of either a pronunciation dictionary for converting source words into phonetic symbols or manually assigned phonetic similarity scores between source and target words. The parameters of the proposed model are automatically learned from a bilingual proper name list. The experimental results for back-transliteration indicate that the proposed method provides significant improvement over previous work.

1 Introduction

Machine transliteration is very important for research and applications in natural language processing, such as machine translation (MT), cross-language information retrieval (CLIR), and bilingual lexicon construction. Proper nouns are often domain specific and frequently created. It is difficult to handle transliteration using existing bilingual dictionaries. Unfamiliar personal names, place names, and technical terms are especially difficult for human translators to transliterate correctly. In CLIR, the accuracy of transliteration greatly affects the retrieval performance.

Recent studies have made great strides toward machine transliteration for many language pairs, such as English/Arabic (Stalls and Knight, 1998; Al-Onaizan and Knight, 2002), English/Chinese (Chen et al., 1998; Wan and Verspoor, 1998; Lin and Chen, 2002), English/Japanese (Knight and Graehl, 1998), and English/Korean (Lee and Choi, 1997; Oh and Choi, 2002). Machine transliteration is classified into two types based on transliteration direction. Transliteration, forward-direction, is the process that converts an original word in the source language into an approximate phonetic equivalent word in the target language, whereas back-transliteration, backward-direction, is the reverse process that converts the transliterated word back into its original word. Most of the previous approaches require a pronunciation dictionary to convert a source word into its corresponding pronunciation sequence. Words with unknown pronunciations may cause problem for transliteration. In addition, using a language-dependent penalty function to measure the similarity between a source word and corresponding transliteration or using handcraft heuristic mapping rules to deal with transliteration may lead to problems when porting to other language pairs.

In this paper, we focus on Chinese-to-English back-transliteration. The proposed framework requires no conversion of source words into phonetic symbols. The model is trained automatically on a bilingual proper name list.

The remainder of the paper is organized as follows: Section 2 presents the proposed statistical transliteration model (STM) and describes the model parameters. In Section 3, we describe the framework to deal with back-transliteration. Experimental setup and the results of the evaluation are presented in Section 4. Concluding remarks are made in Section 5.

2 Statistical Transliteration Model

One can consider machine transliteration as a noisy channel. Under the noisy channel model, the back-transliteration problem is to find the most probable word E from the given transliteration C. Let $P(E)$ be the probability of a word E, then, for a given transliteration C, the back-transliteration

probability of a word E can be written as $P(E|C)$. Since $P(C)$ is constant for the given C, by Bayes' rule, the transliteration problem can be written as follows:

$$\hat{E} = \arg\max_E P(E \mid C) = \arg\max_E \frac{P(E)P(C \mid E)}{P(C)} = \arg\max_E P(E)P(C \mid E), \quad (1)$$

where $\hat{E}$ is the most likely to the word E for the given C, $P(E)$ is the language model, and $P(C|E)$ is the transliteration model.

For the rest of the paper, we assume that E is written in English, while C is written in Chinese. Since Chinese and English are two totally different languages, there is no simple or direct way of mapping and comparison. One feasible solution is to adopt a Chinese romanization system[1] to represent the pronunciation of each Chinese character.

The language model gives the prior probability $P(E)$ which can be modeled using maximum likelihood estimation. As for the transliteration model $P(C|E)$, we can approximate it by decomposing E and romanization of C into transliteration units (TUs)[2]. To illustrate how the approach works, take the example of an English name, "Abe", which can be segmented into three TUs and aligned with the romanized transliteration. Assuming that the word is segmented into "A-b-e", then a possible alignment with the Chinese transliteration "艾貝 (Aipei)" is depicted in Figure 1.

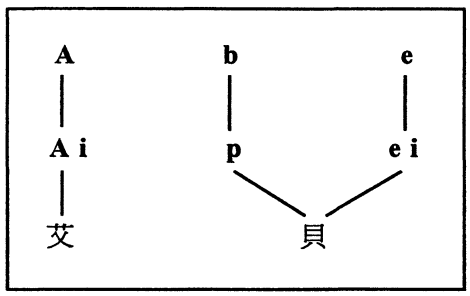

Figure 1. Alignment between English and Chinese romanized character sequences.

Given a specified source character sequence, E, a romanized target character sequence C is the transliteration of E with probability $P(C|E)$. The goal of back-transliteration is to decode the character string E, based on the romanized character sequence C, so that the decoded string $\hat{E}$ has the maximum a posteriori (MAP) probability, i.e.,

$$\hat{E} = \arg\max_E P(E \mid C)P(E). \quad (2)$$

A word E with l characters and a word C with n characters are denoted by e_1^l and c_1^n, respectively. Assume that the number of aligned TUs in (E, C) is N, and let $M = \{m_1, m_2, ..., m_N\}$ be an alignment candidate, where m_j is the *match type* of the j-th TU. The match type is defined as a pair of TU lengths for the two languages. For instance, in the case of (Abe, 艾貝), N is 3, and M is {1-2, 1-1, 1-2}. We write E and C as follows:

---

[1] Ref. sites: "http://www.romanization.com/index.html" and "http://www.edepot.com/taoroman.html".

[2] Transliteration unit is defined as sequence of characters transliterated as a base unit. For English, a TU can be a monograph, a digraph, or a trigraph (Wells, 2001). For Chinese, a TU can be a syllable initial, a syllable final, or a syllable (Chao, 1968) represented by corresponding romanized characters.

$$\begin{cases} E = e_1^l = u_1, u_2, ..., u_N \\ C = c_1^n = v_1, v_2, ..., v_N \end{cases}, \tag{3}$$

where u_i and v_j are the i-th TU of E and the j-th TU of C, respectively.
Then the probability of C given E, $P(C|E)$, is formulated as follows:

$$\begin{aligned} P(C|E) &= \sum_M P(C,M|E) = \sum_M P(C|M,E)P(M|E) \\ &\approx \max_M P(C|M,E)P(M|E) \approx \max_M P(C|M,E)P(M) \end{aligned} \tag{4}$$

We approximate $P(C|E)P(M)$ as follows:

$$\begin{aligned} P(C|M,E)P(M) &= P(v_1^N|u_1^N)P(m_1,m_2,...,m_N) \\ &\approx \prod_{i=1}^{N} P(v_i|u_i)P(m_i). \end{aligned} \tag{5}$$

Therefore, we have

$$\log P(C|E) \approx \max_M \sum_{i=1}^{N} (\log P(v_i|u_i) + \log P(m_i)). \tag{6}$$

Then, for a given C, the best source string $\hat{E}$ can be efficiently obtained by using a dynamic programming algorithm. Using the Expectation Maximization (EM) algorithm (Dempster et al., 1977) with Viterbi decoding (Forney, 1973), we adopt the iterative parameter estimation procedure to solve the maximum likelihood estimation (MLE) problem. For more details, please refer to Lee and Chang (2003).

3 Back-transliteration

The proposed transliteration model can be applied to back-transliteration. The complexity of the task increases for language pairs with different sound systems, such as Chinese/English, Japanese/English, and Arabic/English.

3.1 Similarity-based Framework

There are several approaches to back-transliteration, such as the generative framework, similarity-based framework, and rule-based framework. As stated by Lin and Chen (2002), the similarity-based approach works better because it directly addresses the problem of similarity measurement between the source word and the target word. Therefore, we use the similarity-based approach to model the task of back-transliteration.

Under the similarity-based framework (Lin and Chen, 2002), given a transliterated word, a set of source words was compared with it, and then ranked by similarity scores. The most similar word is chosen as the answer to the back-transliteration problem. However, in order to measure similarity at the grapheme level (Lin and Chen 2002), Chinese words and English words are first converted into phonemes and then represented according to the International Phonetic Alphabet.

One serious limitation of the scheme proposed by Lin and Chen (2002) is that many proper names are not covered by existing pronunciation dictionaries. The transliteration approach we proposed measures directly the similarity between the source word and the target word at the grapheme level. No conversion of source words into phonetic symbols is needed in our approach, as shown in Figure 2. First, the given Chinese word is romanized by simply table lookup. Then, the similarity between the romanized Chinese and each of the members of a pre-collected set of English proper nouns is calculated by using our proposed transliteration model to produce a list of ranked candidates.

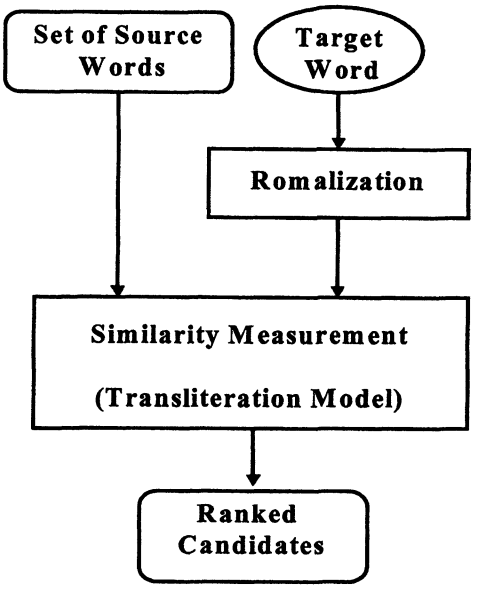

Figure 2. Direct similarity measurement without phonetic conversion.

3.2 An Example

For example, given a Chinese transliterated word "柯爾品", the goal is to find out the original source word "Kolpin". We first romanize "柯爾品" into "Koerhpin", then the proposed model is employed to measure the similarities between the romanized word and each of the members of a pre-collected set of English proper nouns. In our experiments, the top 4 candidates are "Kolpin", "Kleppner", "Charley", and "Columbine", respectively.

For simplicity, we only show the TU alignments of each source-target word pairs in Figure 3. In this case, the correct answer "Kolpin", the most likely source word of the Chinese transliterated word "柯爾品" is chosen as the top 1 candidate. The number of aligned TUs for (Kolpin, Koerhpin) is 6. The match types of this alignment are {1-1, 1-1, 1-3, 1-1, 1-1, 1-1}.

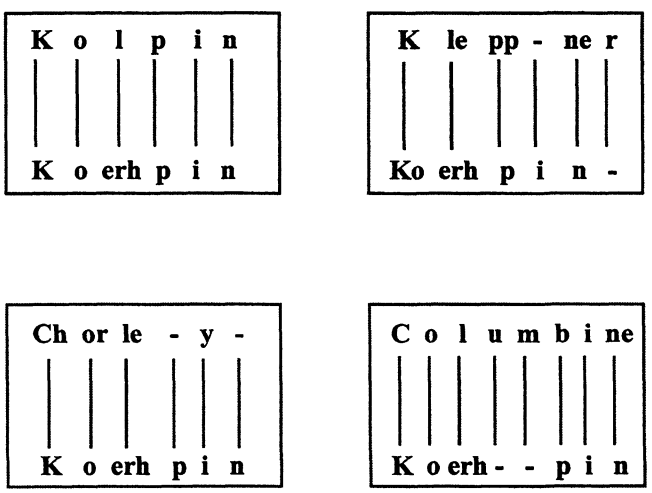

Figure 3. TU alignment of "柯爾品 (Koerhpin)" and corresponding source candidate words.

4 Experiments

In this section, we focus on the setup for the experiments and a performance evaluation of the proposed model applied to back-transliteration.

4.1 Experimental Setup

The corpus *T0* for training consists of 2,430 pairs of English and their transliterated Chinese names. To evaluate the performance, 150 unseen personal name pairs were collected to form the test set *T1*. A validation set *T2*, consisting of another 150 unseen personal name pairs, was collected for analyzing the learning curve. For each transliterated word in *T1* (or *T2*), a set of 1,557 source words was compared with it. Table 1 shows some samples of *T0*.

Table 1. Some samples from the training set *T0*.

Source word	Target word	Source word	Target word
Abe	阿貝	Agatha	阿佳莎
Abbey	阿比	Acton	阿克頓
Abbot	阿伯特	Arkwright	阿克賴特
Archer	阿徹	Arabella	阿拉蓓拉
Adolf	阿道夫	Alaric	阿拉里克
Adolphus	阿道弗斯	Alasdair	阿拉斯代爾
Adela	阿德拉	Alastair	阿拉斯泰爾
Adelaide	阿德萊德	Alethea	阿蕾西
Arden	阿登	Alonzo	阿朗索
Albert	阿爾伯特	Ariadne	阿莉雅德妮
Alfonso	阿爾方索	Allegra	阿莉葛拉
Alfie	阿爾菲	Alister	阿利斯特
Alf	阿爾夫	Allie	阿莉
Algy	阿爾吉	Arlene	阿琳
Algernon	阿爾傑農	Alan	阿倫
Alma	阿爾瑪	Aloys	阿洛伊斯
Almeric	阿爾梅里克	Aloysius	阿洛伊修斯

The performance is evaluated by rates of the Average Rank (*AR*) and the Average Reciprocal Rank (*ARR*) following Voorhees and Tice (2000).

$$AR = \frac{1}{N}\sum_{i=1}^{N} R(i), \tag{7}$$

$$ARR = \frac{1}{N}\sum_{i=1}^{N} 1/R(i), \tag{8}$$

where N is the number of testing data, and $R(i)$ is the rank of the i-th testing data.

4.2 Experimental Results and Discussion

In Figure 4, we show the rate of *AR* for *T1* and *T2* according to the size of *T0*. Based on *AR*, as shown in Figure 4, the performance began to converged when the size was around 800 English-Chinese name pairs.

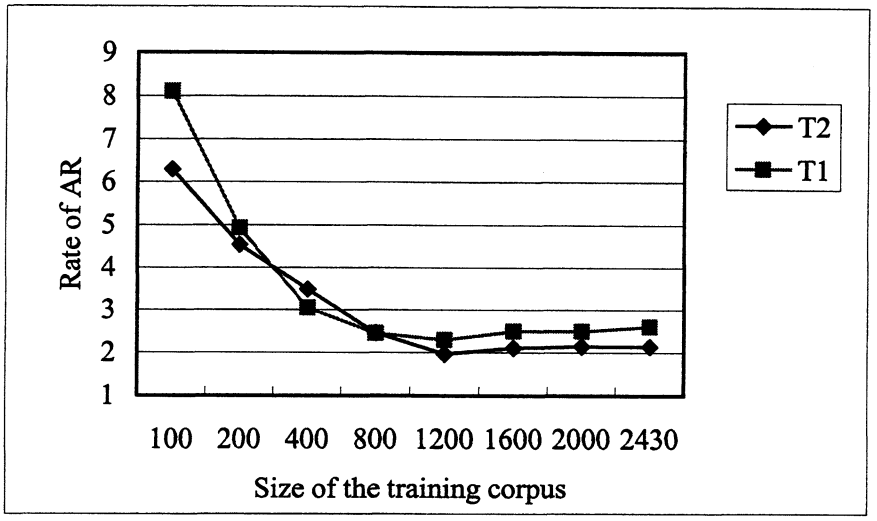

Figure 4. Rates of *AR* for *T1* and *T2* according to the size of *T0*.

A baseline method was first established by the Edit Distance Algorithm (EDA)[3] (Hall and Dowling, 1980), applied on characters to transfer one word into another word. The second baseline model, Weighted Edit Distance Algorithm (WEDA), enhanced the EDA with two more features. First, the first TU of each romanized Chinese character is considered more important than the others; we put a weighted penalty on the first TU of each romanized Chinese character. Second, the length of TUs is allowed to be more than 1 to model more pronunciation rules, for example, ("bb", "p"), ("ph", "f"), and ("wh", "hu").

The results with the edit distance measure approaches, EDA and WEDA, are shown in the first two columns of Table 2. The result of our proposed model, STM, is shown in row 3 of Table 2. Though WEDA appears to be better than EDA, our method STM can further improve the performance significantly. According to the experimental results in Table 2, our methods, STM, is quite efficient. The experimental results show that our methods have significantly more discriminative power than the methods of EDA and WEDA based on both *AR* and *ARR*.

Table 2. The experimental results produced by the proposed method for *T1*.

Method	*AR*	*ARR*
EDA	46.59	0.4973
WEDA	21.45	0.6780
STM	2.30	0.8347

[3] In the Wade-Giles romanization system, there is no distinction between the TUs "p" and "b," which are both represented by "p." The same also applies to the other TU pairs, such as ("c", "k"), ("d", "t"), ("g", "k"), ("r", "l"), and ("v", "f"). The EDA approach may encounter problems when such variations are encountered. Therefore, we viewed these pairs as equivalent ones individually in our experiments.

Figure 5 shows the performance achieved based on the rank distributions of the correct candidates. In Table 3 and Table 4, we show some examples with the top 5 candidates for the EDA and WEDA, respectively. All of the source words shown in Table 3 and Table 4 were correctly decoded using the proposed method, STM. For example, as shown in Table 3, the ranks of the decoded source words "Bernadine", "Jeannette", "Barnabas", "Bennie", "Nannie", and "Bennett" are 1, 1, 3, 3, 3, and 7, respectively, for the transliterated word "班奈特 (Pannaite)". Note that the edit distance was normalized by the length of each source word candidate.

Table 5 shows the candidate lists obtained using the proposed model, given the same transliterated words listed in Table 3 and Table 4. It is worth noting that the proposed model produced more likely candidates than the EDA and WEDA approaches did, based on the phonetic similarities between candidates and given words. In other words, STM captured more phonetic information in the back-transliteration process than EDA and WEDA did. There is one further point which should not be ignored. EDA and WEDA were less discriminative because they produced source word candidates with the same rank order.

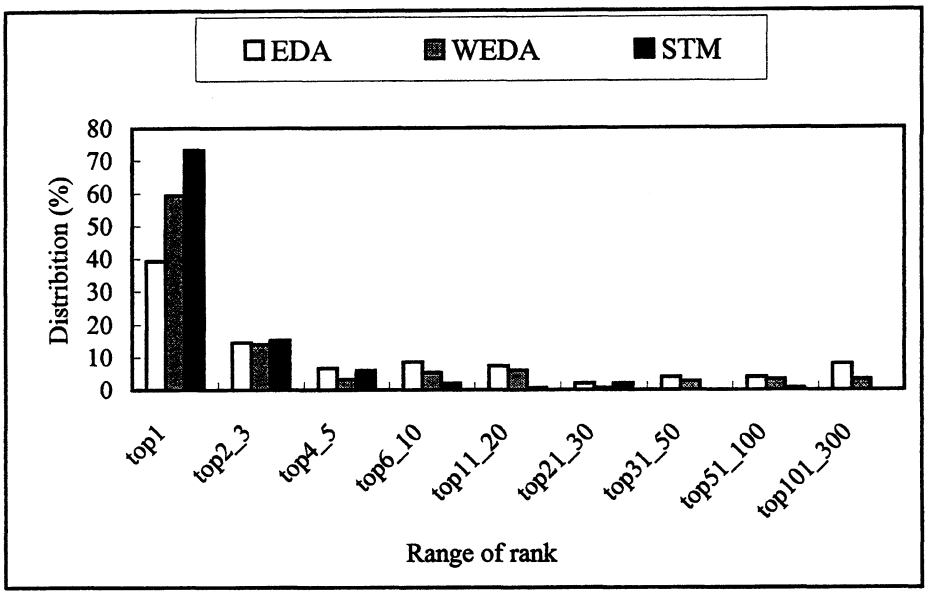

Figure 5 Rank distributions.

Table 3. Some examples of back-transliteration with the top 5 candidates generated by the EDA approach.
(The numbers enclosed in parentheses "()" indicate the ranks of the decoded source words.)

Target word	Source word	Top 1	Top 2	Top 3	Top 4	Top 5
班奈特 Pannaite	Bennett (7)	Bernadine (1)	Jeannette (1)	Barnabas (3)	Bennie (3)	Nannie (3)
凱斯 Kaissu	Cayce (215)	Crispus (1)	Laxson (1)	Caesar (3)	Cissie (3)	Gascon (3)
庫里 Kuli	Cooley (17)	Kolin (1)	Kurt (2)	Lori (2)	Gore (2)	Kolpin (5)
赫勒 Hole	Heller (16)	Hale (1)	Home (1)	More (1)	Gore (1)	Holmes (5)
傑夫里茲 Chiehfulitzu	Jefferies (20)	Chesterfield (1)	Chevalier (2)	Siegfried (2)	Theodoric (2)	Alastairfitter (5)

Table 4. Some examples of back-transliteration with the top 5 candidates generated by the WEDA approach.
(The numbers enclosed in parentheses "()" indicate the ranks of the decoded source words.)

Target word	Source word	Top 1	Top 2	Top 3	Top 4	Top 5
班奈特 Pannaite	Bennett (4)	Benita (1)	Bonita (1)	Bennie (3)	Bennett (4)	Anita (5)
凱斯 Kaissu	Cayce (127)	Wise (1)	Cissie (2)	Gareth (2)	Gasser (2)	Krause (2)
庫里 Kuli	Cooley (2)	Curry (1)	Cooley (2)	Carrie (3)	Colley (3)	Garry (5)
赫勒 Hole	Heller (10)	Hurley (1)	Holly (2)	Hale (3)	Noele (4)	Charley (5)
傑夫里茲 Chiehfulitzu	Jefferies (2)	Kiesslingcooper (1)	Jefferies (2)	Siegfried (2)	Theodoric (2)	Katherine (5)

Table 5. Some examples of back-transliteration with the top 5 candidates generated by the STM approach.
(The numbers enclosed in parentheses "()" indicate the ranks of the decoded source words.)

Target word	Source word	Top 1	Top 2	Top 3	Top 4	Top 5
班奈特 Pannaite	Bennett (1)	Bennett (1)	Jeannette (2)	Barney (3)	Bonita (4)	Bennie (5)
凱斯 Kaissu	Cayce (1)	Cayce (1)	Gareth (2)	Chase (3)	Gasser (4)	Carnes (5)
庫里 Kuli	Cooley (1)	Cooley (1)	Chorley (2)	Colley (3)	Curry (4)	Cowley (5)
赫勒 Hole	Heller (1)	Heller (1)	Holly (2)	Harry (3)	Hurley (4)	Hale (5)
傑夫里茲 Chiehfulitzu	Jefferies (1)	Jefferies (1)	Geoffrey (2)	Jeffrey (3)	Siegfried (4)	Chevalier (5)

For the sake of more natural Mandarin pronunciation, the target word is transliterated from the corresponding source word with insertions, deletions, or substitutions of TUs during transliteration. These exceptions are the main cause of failure during back-transliteration in our method. For example, these test data, (Aguirre, 阿基瑞 "Achijui"), (Bogard, 波嘉 "Pochia"), (Descartes, 笛卡兒 "Tikaerh"), and (Lang, 蘭恩 "Lanen") were not correctly ranked as top 1 candidates by the proposed method.

5 Conclusion

An automatic learning approach to the machine transliteration problem is presented in this paper. We rely on statistics gathered from a bilingual name list. The experimental results show that the proposed method significantly outperforms previous work. Furthermore, the proposed model is also applicable to the ongoing task of extracting proper names and transliterations.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 400-408.

Hsin-Hsi Chen, Sheng-Jie Huang, Yung-Wei Ding, and Shih-Chung Tsai. 1998. Proper name translation in cross-language information retrieval. In *Proceedings of 17th COLING and 36th ACL*, pages 232-236.

Yuen Ren Chao. 1968. A Grammar of spoken Chinese. Berkeley, University of California Press.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1-38.

G. D. Forney. 1973. The Viterbi algorithm. *Proceedings of IEEE*, 61:268-278, March.

Patrick A.V. Hall and Geoff R. Dowling. 1980. Approximate String Matching. *ACM Computing Surveys*, 12:381-402.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599-612.

Chun-Jen Lee and Jason S. Chang. 2003. Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 96-103, Edmonton, Canada.

Jae Sung Lee and Key-Sun Choi. 1997. A statistical method to generate various foreign word transliterations in multilingual information retrieval system. In *Proceedings of the 2nd International Workshop on Information Retrieval with Asian Languages (IRAL'97)*, pages 123-128, Tsukuba, Japan.

Wei-Hao Lin and Hsin-Hsi Chen. 2002. Backward transliteration by learning phonetic similarity. In *CoNLL-2002, Sixth Conference on Natural Language Learning*, Taipei, Taiwan.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in Arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

Ellen M. Voorhees and Dawn M. Tice. 2000. The trec-8 question answering track report. In *English Text Retrieval Conference (TREC-8)*.

Stephen Wan and Cornelia Maria Verspoor. 1998. Automatic English-Chinese name transliteration for development of multilingual resources. In *Proceedings of 17th COLING and 36th ACL*, pages 1352-1356.

J. C. Wells. 2001. *Longman Pronunciation Dictionary* (*New Edition*), Addison Wesley Longman, Inc.