# Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts using a Statistical Machine Transliteration Model

Chun-Jen Lee[1, 2]
[1] Telecommunication Labs.
Chunghwa Telecom Co., Ltd.
Chungli, Taiwan, R.O.C.
cjlee@cht.com.tw

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

## Abstract

This paper presents a framework for extracting English and Chinese transliterated word pairs from parallel texts. The approach is based on the statistical machine transliteration model to exploit the phonetic similarities between English words and corresponding Chinese transliterations. For a given proper noun in English, the proposed method extracts the corresponding transliterated word from the aligned text in Chinese. Under the proposed approach, the parameters of the model are automatically learned from a bilingual proper name list. Experimental results show that the average rates of word and character precision are 86.0% and 94.4%, respectively. The rates can be further improved with the addition of simple linguistic processing.

## 1 Introduction

Automatic bilingual lexicon construction based on bilingual corpora has become an important first step for many studies and applications of natural language processing (NLP), such as machine translation (MT), cross-language information retrieval (CLIR), and bilingual text alignment. As noted in Tsuji (2002), many previous methods (Dagan et al., 1993; Kupiec, 1993; Wu and Xia, 1994; Melamed, 1996; Smadja et al., 1996) deal with this problem based on frequency of words appearing in the corpora, which can not be effectively applied to low-frequency words, such as transliterated words. These transliterated words are often domain-specific and created frequently. Many of them are not found in existing bilingual dictionaries. Thus, it is difficult to handle transliteration only via simple dictionary lookup. For CLIR, the accuracy of transliteration highly affects the performance of retrieval.

In this paper, we present a framework of acquisition for English and Chinese transliterated word pairs based on the proposed statistical machine transliteration model. Recently, much research has been done on machine transliteration for many language pairs, such as English/Arabic (Al-Onaizan and Knight, 2002), English/Chinese (Chen et al., 1998; Lin and Chen, 2002; Wan and Verspoor, 1998), English/Japanese (Knight and Graehl, 1998), and English/Korean (Lee and Choi, 1997; Oh and Choi, 2002). Most previous approaches to machine transliteration have focused on the use of a pronunciation dictionary for converting source words into phonetic symbols, a manually assigned scoring matrix for measuring phonetic similarities between source and target words, or a method based on heuristic rules for source-to-target word transliteration. However, words with unknown pronunciations may cause problems for transliteration. In addition, using either a language-dependent penalty function to measure the similarity between bilingual word pairs, or handcrafted heuristic mapping rules for transliteration may lead to problems when porting to other language pairs.

The proposed method in this paper requires no conversion of source words into phonetic symbols. The model is trained automatically on a bilingual proper name list via unsupervised learning.

The remainder of the paper is organized as follows: Section 2 gives an overview of machine transliteration and describes the proposed model. Section 3 describes how to apply the model for extraction of transliterated target words from parallel texts. Experimental setup and quantitative assessment of performance are presented in Section 4. Concluding remarks are made in Section 5.

## 2 Statistical Machine Transliteration Model

### 2.1 Overview of the Noisy Channel Model

Machine transliteration can be regarded as a noisy channel, as illustrated in Figure 1. Briefly, the language model generates a source word $E$ and the transliteration model converts the word $E$ to a target transliteration $C$. Then, the channel decoder is used to find the word $\hat{E}$ that is the most likely to the word $E$ that gives rise to the transliteration $C$.

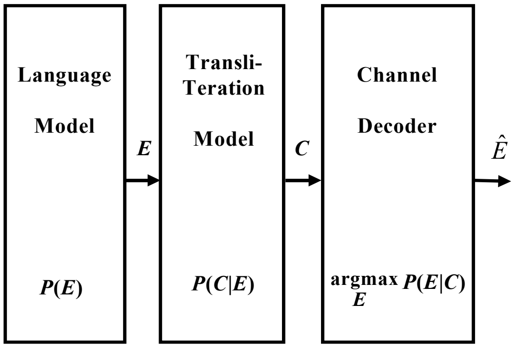


Figure 1. The noisy channel model in machine transliteration.

Under the noisy channel model, the back-transliteration problem is to find out the most probable word $E$, given transliteration $C$. Letting $P(E)$ 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 $P(E|C)$. By Bayes' rule, the transliteration problem can be written as follows:

$$\hat{E} = \underset{E}{\arg\max} P(E \mid C) = \underset{E}{\arg\max} \frac{P(E)P(C \mid E)}{P(C)}. \tag{1}$$

Since $P(C)$ is constant for the given $C$, we can rewrite Eq. (1) as follows:

$$\hat{E} = \underset{E}{\arg\max}\, P(E)P(C \mid E), \tag{2}$$

The first term, $P(E)$, in Eq. (2) is the language model, the probability of $E$. The second term, $P(C|E)$, in Eq. (2) is the transliteration model, the probability of the transliteration $C$ conditioned on $E$.

Below, we assume that $E$ is written in English, while $C$ is written in Chinese. Since Chinese and English are not in the same language family, 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. Among the many romanization systems for Chinese, Wade-Giles and Pinyin are the most widely used. The Wade-Giles system is commonly used in Taiwan today and has traditionally been popular among Western scholars. For this reason, we use the Wade-Giles system to romanize Chinese characters. However, the proposed approach is equally applicable to other romanization systems.

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 using the transliteration unit (TU), which is a decomposition of $E$ and $C$. TU is defined as a 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. To illustrate how this approach works, take the example of an English name, "Smith", which can be segmented into four TUs and aligned with the romanized transliteration. Assuming that the word is segmented into "S-m-i-th", then a possible alignment with the Chinese transliteration "史密斯 (Shihmissu)" is depicted in Figure 2.

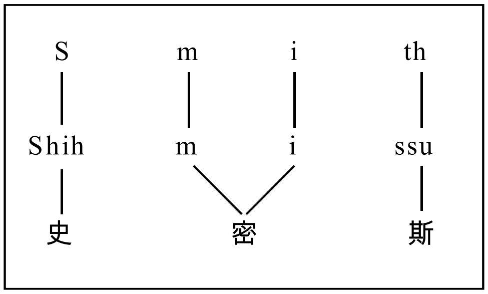


Figure 2. TU alignment between English and Chinese romanized character sequences.

### 2.2 Formal Description: Statistical Transliteration Model (STM)

A word $E$ with $l$ characters and a romanized word $C$ with $n$ characters are denoted by $e_1^l$ and $c_1^n$, respectively. Assume that the number of aligned TUs for ($E$, $C$) 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 (Smith, Shihmissu), $N$ is 4, and $M$ is {1-4, 1-1, 1-1, 2-3}. We write $E$ and $C$ as follows:

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

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:

$$P(C \mid E) = \sum_M P(C, M \mid E) = \sum_M P(C \mid M, E)P(M \mid E). \tag{4}$$

To reduce computational complexity, one alternative approach is to modify the summation criterion in Eq. (4) into maximization. Therefore, we can approximate $P(C|E)$ as follows:

$$\begin{aligned} P(C \mid E) &\approx \max_M P(C \mid M, E)P(M \mid E) \\ &\approx \max_M P(C \mid M, E)P(M). \end{aligned} \tag{5}$$

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

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

$$P(C\mid M,E)P(M) = P(v_1^N \mid u_1^N)P(m_1, m_2, ..., m_N)$$

$$\approx \prod_{i=1}^{N} P(v_i \mid u_i)P(m_i). \qquad (6)$$

Therefore, we have

$$\log P(C\mid E) \approx \max_{M} \sum_{i=1}^{N} \left(\log P(v_i \mid u_i) + \log P(m_i)\right).$$

(7)

Let $S(i,j)$ be the maximum accumulated log probability between the first *i* characters of *E* and the first *j* characters of *C*. Then, $\log P(C\mid E) = S(l,n)$, the maximum accumulated log probability among all possible alignment paths of *E* with length *l* and *C* with length *n*, can be computed using a dynamic programming (DP) strategy, as shown in the following:

**Step 1 (Initialization):**

$$S(0,0) = 0$$

(8)

**Step 2 (Recursion):**

$$S(i,j) = \max_{h,k} S(i-h, j-k) + \log P(c_{j-k}^{j} \mid e_{i-h}^{i}) + \log P(h,k)$$

$$0 \le i \le l, \quad 0 \le j \le n.$$

(9)

**Step 3 (Termination):**

$$S(l,n) = \max_{h,k} S(l-h, n-k) + \log P(c_{n-k}^{n} \mid e_{l-h}^{l}) + \log P(h,k)$$

(10)

where $P(h,k)$ is defined as the probability of the match type "*h-k*".

### 2.3 Estimation of Model Parameters

To describe the iterative procedure for re-estimation of probabilities of $P(v_j \mid u_i)$ and $P(m_i)$, we first define the following functions:

$count(u_i, v_j)$ = *the number of occurrences of aligned pair* $u_i$ *and* $v_i$ *in the training set.*

$count(u_i)$ = *the number of occurrences of* $u_i$ *in the training set.*

$count(h,k)$ = *the total number of occurrences of* $u_i$ *with length h aligned with* $v_j$ *with length k in the training set.*

Therefore, the translation probability $P(v_j \mid u_i)$ can be approximated as follows:

$$P(v_j \mid u_i) = \frac{count(u_i, v_j)}{count(u_i)}.$$

(11)

The probability of the match type, $P(h,k)$, can be estimated as follows:

$$P(h,k) = \frac{count(h,k)}{\sum_i \sum_j count(i,j)}.$$

(12)

For the reason that $count(u_i, v_j)$ is unknown in the beginning , a reasonable initial estimate of the parameters of the translation model is to constrain the TU alignments of a word pair (*E*, *C*) within a position distance $\delta$ (Lee and Choi, 1997). Assume that $u_i = e_p^{p+h-1}$ and $v_j = c_q^{q+k-1}$, and $d_\delta(u_i, v_j)$ is the allowable position distance within $\delta$ for the aligned pair ($u_i$, $v_i$). $d_\delta(u_i, v_j)$ is defined as follows:

$$d_\delta(u_i, v_j) = \begin{cases} \left| p - \dfrac{q \times l}{n} \right| < \delta, & and \\ \left| (p+h-1) - \dfrac{(q+k-1) \times l}{n} \right| < \delta \end{cases}, \qquad (13)$$

where *l* and *n* are the length of the source word *E* and the target word *C*, respectively.

To accelerate the convergence of EM training and reduce the noisy TU aligned pairs ($u_i$, $v_j$), we restrict the combination of TU pairs to limited patterns. Consonant TU pairs only with same or similar phonemes are allowed to be matched together. An English consonant is also allowed to matching with a Chinese syllable beginning with same or similar phonemes. An English semivowel TU can either be matched with a Chinese consonant or a vowel with same or similar phonemes, or be matched with a Chinese syllable beginning with same or similar phonemes.

As for the probability of match type, $P(h,k)$, it is set to uniform distribution in the initialization phase, shown as follows:

$$P(h,k) = \frac{1}{T}, \qquad (14)$$

where *T* is the total number of match types allowed.

Based on the Expectation Maximization (EM) algorithm (Dempster et al., 1977) with Viterbi decoding (Forney, 1973), the iterative parameter estimation procedure is described as follows:

**Step 1 (Initialization):**

Use Eq. (13) to generate likely TU alignment pairs. Calculate the initial model parameters,

$P(v_j \mid u_i)$ and $P(h,k)$, using Eq. (11) and Eq. (12).

**Step 2 (Expection):**

Based on current model parameters, find the best Viterbi path for each *E* and *C* word pair in the training set.

**Step 3 (Maximization):**

Based on all the TU alignment pairs obtained from Step 2, calculate the new model parameters using Eqs. (11) and (12). Replace the model parameters with the new model parameters. If it reaches a stopping criterion or a predefined iteration numbers, then stop the training procedure. Otherwise, go back to Step 2.

## 3 Extraction of Transliteration from Parallel Text

The task of machine transliteration is useful for many NLP applications, and one interesting related problem is how to find the corresponding transliteration for a given source word in a parallel corpus. We will describe how to apply the proposed model for such a task.

For that purpose, a sentence alignment procedure is applied first to align parallel texts at the sentence level. Then, we use a tagger to identify proper nouns in the source text. After that, the model is applied to isolate the transliteration in the target text. In general, the proposed transliteration model could be further augmented with linguistic processing, which will be described in more details in the next subsection. The overall process is summarized in Figure 3.

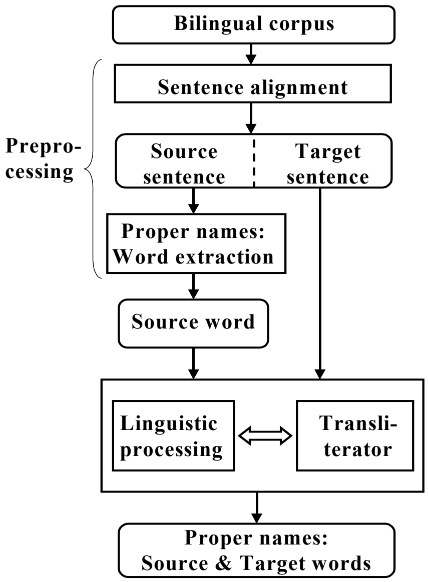


Figure 3. The overall process for the extraction of transliteration from parallel text.

An excerpt from the magazine *Scientific American* (Cibelli et al., 2002) is illustrated as follows:

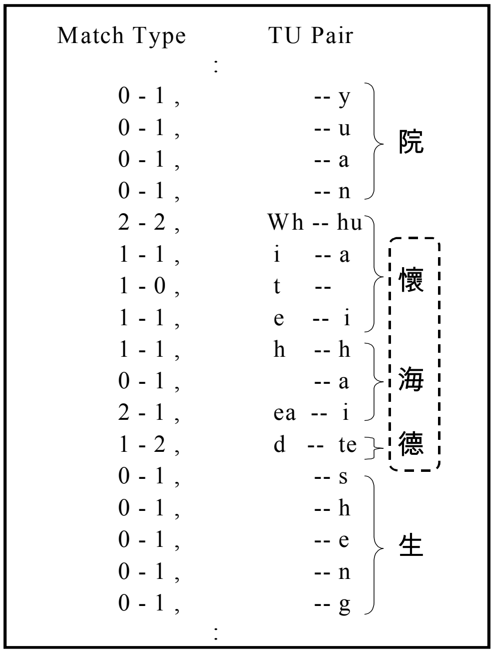
**Source sentence:**

"Rudolf Jaenisch, a cloning expert at the Whitehead Institute for Biomedical Research at the Massachusetts Institute of Technology, concurred:"

**Target sentence:**

"麻省理工學院懷海德生物醫學研究院的複製專家傑尼西說："

In the above excerpt, three English proper nouns "Jaenisch", "Whitehead", and "Massachusetts" are identified by a tagger. Utilizing Eqs. (7) and the DP approach formulated by Eqs. (8)-(10), we found the target word "huaihaite (懷海德)" most likely corresponding to "Whitehead". In order to retrieve the transliteration for a given proper noun, we need to keep track of the optimal TU decoding sequence associated with the given Chinese term for each word pair under the proposed method. The aligned TUs can be easily obtained via backtracking the best Viterbi path (Manning and Schutze, 1999). For the example mentioned above, the alignments of the TU matching pairs via the Viterbi backtracking path are illustrated in Figure 4.



| Match Type | TU Pair | |
|---|---|---|
| | : | |
| 0 - 1 , | -- y | 院 |
| 0 - 1 , | -- u | |
| 0 - 1 , | -- a | |
| 0 - 1 , | -- n | |
| 2 - 2 , | Wh -- hu | 懷 |
| 1 - 1 , | i -- a | |
| 1 - 0 , | t -- | |
| 1 - 1 , | e -- i | |
| 1 - 1 , | h -- h | 海 |
| 0 - 1 , | -- a | |
| 2 - 1 , | ea -- i | |
| 1 - 2 , | d -- te | 德 |
| 0 - 1 , | -- s | 生 |
| 0 - 1 , | -- h | |
| 0 - 1 , | -- e | |
| 0 - 1 , | -- n | |
| 0 - 1 , | -- g | |
| | : | |

Figure 4. The alignments of the TU matching pairs via the Viterbi backtracking path.

### 3.1 Linguistic Processing

Some language-dependent knowledge can be integrated to further improve the performance, especially when we focus on specific language pairs.

**Linguistic Processing Rule 1 (R1):**
Some source words have both transliteration and translation, which are equally acceptable and can be used interchangeably. For example, the source word “England” is translated into “英國 (Yingkou)” and transliterated into “英格蘭 (Yingkolan)”, respectively, as shown in Figure 5. Since the proposed model is designed specifically for transliteration, such cases may cause problems. One way to overcome this limitation is to handle those cases by using a list of commonly used proper names and translations.

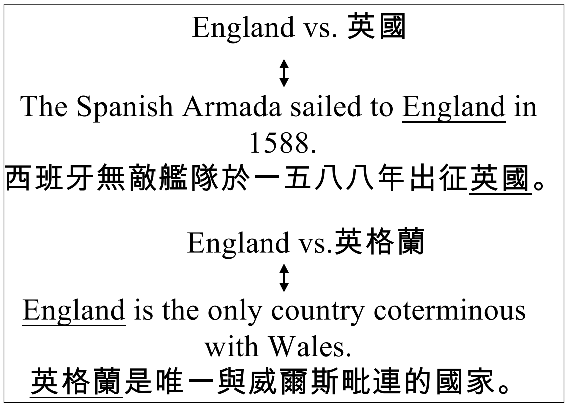

England vs. 英國
The Spanish Armada sailed to England in 1588.
西班牙無敵艦隊於一五八八年出征英國。
England vs.英格蘭
England is the only country coterminous with Wales.
英格蘭是唯一與威爾斯毗連的國家。

Figure 5. Examples of mixed usages of translation and transliteration.

**Linguistic Processing Rule 2 (R2):**
From error analysis of the aligned results of the training set, the proposed approach suffers from the fluid TUs, such as “t”, “d”, “tt”, “dd”, “te”, and “de”. Sometimes they are omitted in transliteration, and sometimes they are transliterated as a Chinese character. For instance, “d” is usually transliterated into “特”, “得”, or “德” corresponding to Chinese TU of “te”. The English TU “d” is transliterated as “德” in (Clifford, 克利福德), but left out in (Radford, 雷德福). In the example shown in Figure 6, “David (大衛)” is mistakenly matched up with “大衛的”.

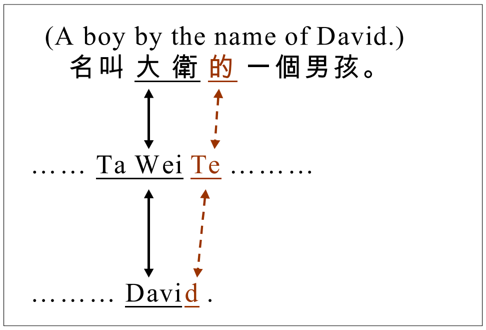


Figure 6. Example of the transliterated word extraction for “David”.

However, that problem caused by fluid TUs can be partly overcome by adding more linguistic constraints in the post-processing phase. We calculate the Chinese character distributions of proper nouns from a bilingual proper name list. A small set of Chinese characters is often used for transliteration. Therefore, it is possible to improve the performance by pruning extra tailing characters, which do not belong to the transliterated character set, from the transliteration candidates. For instance, the probability of “的, 去, 說, 是, 有” being used in transliteration is very low. So correct transliteration “大衛” for the source word “David” could be extracted by removing the character “的”.

### 3.2 Working Flow by Integrating Linguistic and Statistical Information

Combining the linguistic processing and transliteration model, we present the algorithm for transliteration extraction as follows:

**Step 1:** Look up the translation list as stated in R1. If the translation of a source word appears in both the entry of the translation list and the aligned target sentence (or paragraph), then pick the translation as the target word. Otherwise, go to Step 2.

**Step 2:** Pass the source word and its aligned target sentence (or paragraph) through the proposed model to extract the target word.

**Step 3:** Apply linguistic processing R2 to remove superfluous tailing characters in the target word.

After the above processing, the performance of source-target word extraction is significantly improved over the previous experiment.

# 4 Experiments

In this section, we focus on the setup of experiments and performance evaluation for the proposed model.

## 4.1 Experimental Setup

The corpus *T0* for training consists of 2,430 pairs of English names together with their Chinese transliterations. Two experiments are conducted. In the first experiment, we analyze the convergence characteristics of this model training based on a similarity-based framework (Chen et al., 1998; Lin and Chen, 2002). A validation set *T1*, consisting of 150 unseen person name pairs, was collected from Sinorama Magazine (Sinorama, 2002). For each transliterated word in *T1*, a set of 1,557 proper names is used as potential answers. In the second experiment, a parallel corpus *T2* was prepared to evaluate the performance of proposed methods. *T2* consists of 500 bilingual examples from the English-Chinese version of the Longman Dictionary of Contempory English (LDOCE) (Proctor, 1988).

## 4.2 Evaluation Metric

In the first experiment, a set of source words was compared with a given target word, and then was ranked by similarity scores. The source word with the highest similarity score is chosen as the answer to the back-transliteration problem. The performance is evaluated by rates of the Average Rank (*AR*) and the Average Reciprocal Rank (*ARR*) following Voorhees and Tice (2000).

$$AR = \frac{1}{N}\sum_{i=1}^{N} R(i) \quad (15)$$

$$ARR = \frac{1}{N}\sum_{i=1}^{N} 1/R(i) \quad (16)$$

where $N$ is the number of testing data, and $R(i)$ is the rank of the $i$-th testing data. Higher values of *ARR* indicate better performance.

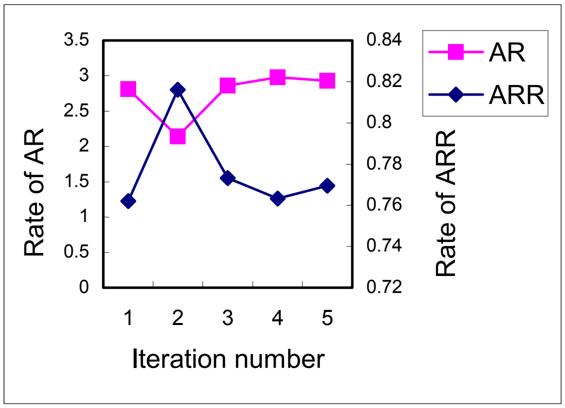


Figure 7. Performance at each iteration on the validation set *T1*.

In Figure 7, we show the rates of *AR* and *ARR* for the validation set *T1* by varying the number of iterations of the EM training algorithm from 1 to 6. We note that the rates become saturated at the 2nd iteration, which indicates the efficiency of the proposed training approach.

As for the second experiment, performance on the extraction of transliterations is evaluated based on precision and recall rates on the word and character level. Since we consider exact one proper name in the source language and one transliteration in the target language at a time. The word recall rates are same as word precision rates:

$$Word\ \ Precision\ \ (WP) = \frac{number\ \ of\ \ correctly\ \ extracted\ \ words}{number\ \ of\ \ correct\ \ words}. \quad (17)$$

The character level recall and precision are as follows:

$$Character\ \ precision\ (CP) = \frac{number\ \ of\ \ correctly\ \ extracted\ \ characters}{number\ \ of\ \ correct\ \ characters}, \quad (18)$$

$$Character\ Recall\ (CR) = \frac{number\ \ of\ \ correctly\ \ extracted\ \ characters}{number\ \ of\ \ correct\ \ characters}. \quad (19)$$

For the purpose of easier evaluation, *T2* was designed to contain exact one proper name in the source language and one transliteration in the target language for each bilingual example. Therefore, if more than one proper name occurs in a bilingual example, we separate them into several testing examples. We also separate a compound proper name in one example into individual names to form multiple examples. For example, in the first case, two proper names "Tchaikovsky" and "Stravinsky" were found in the testing sample "Tchaikovsky and Stravinsky each wrote several famous ballets". In the second case, a compound proper name "Cyril Tourneur" was found in "No one knows who wrote that play, but it is usually ascribed to Cyril Tourneur". However, in the third case, "New York" is transliterated as a whole Chinese word "紐約", so it can not be separated into two words. Therefore, the testing data for the above examples will be semi-automatically constructed. For simplicity, we considered each proper name in the source sentence in turn and determined its corresponding transliteration independently. Table 1 shows some examples of the testing set *T2*.

| |
|---|
| He is a (second) Caesar in speech and leadership.<br>他在演說及領導方面的才能有如凱撒再世. |
| Can you adduce any reason at all for his strange behaviour, Holmes?<br>福爾摩斯, 你能否舉出什麼理由解釋他的古怪行為? |
| They appointed him to catch all the rats in Hamelin.<br>他們指派他捉漢姆林區所有的老鼠. |
| Drink Rossignol, the aristocrat of table wines!<br>喝羅西諾酒吧! 這是餐酒中的上品! |
| Cleopatra was bitten by an asp.<br>克利奧佩特拉女王是被小毒蛇咬死的. |
| Schoenberg used atonality in the music of his middle period.<br>桑伯格在中期用無調性方式作曲. |
| Now that this painting has been authenticated as a Rembrandt, it's worth 10 times as much as I paid for it!<br>由於這幅畫已證實是倫布朗真蹟, 它的時價是我當初買下來時的十倍! |

Table 1. Part of bilingual examples of the testing set *T2*.

In the experiment of transliterated word extraction, the proposed method achieves on average 86.0% word accuracy rate, 94.4% character precision rate, and 96.3% character recall rate, as shown in row 1 of Table 2. The performance can be further improved with a simple statistical and linguistic processing, as shown in Table 2.

| Methods | *WP* | *CP* | *CR* |
|---|---|---|---|
| Baseline | 86.0% | 94.4% | 96.3% |
| Baseline+R1 | 88.6% | 95.4% | 97.7% |
| Baseline+R2 | 90.8% | 97.4% | 95.9% |
| Baseline+R1+R2 | 94.2% | 98.3% | 97.7% |

Table 2. The experimental results of transliterated word extraction for *T2*.

In the baseline model, we find that there are some errors caused by translations which are not strictly transliterated; and there are some source words transferred into target words by means of transliteration and translation mutually. Therefore, R1 can be viewed as the preprocessing to extract transliterated words. Some errors are further eliminated by R2 which considers the usage of the transliterated characters in the target language. In this experiment, we use a transliterated character set of 735 Chinese characters.

## 5 Conclusion

In this paper, we describe a framework to deal with the problem of acquiring English-Chinese bilingual transliterated word pairs from parallel-aligned texts. An unsupervised learning approach to the proposed machine transliteration model is also presented. The proposed approach automatically learned the parameters of the model from a bilingual proper name list. It is not restricted to the availability of pronunciation dictionary in the source language. From the experimental results, it indicates that our methods achieve excellent performance. With the statistical-based characteristic of the proposed model, we plan to extend the experiments to bi-directional transliteration and other different corpora.

## 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.

Dagan, I., Church, K. W., and Gale, W. A. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 1-8, Columbus Ohio.

Jose B. Cibelli, Robert P. Lanza, Michael D. West, and Carol Ezzell. 2002. What Clones? *Scientific American*, Inc., New York, January. http://www.sciam.com.

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.

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

Julian Kupiec. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 40th Annual Conference of the*

*Association for Computational Linguistics (ACL)*, pages 17-22, Columbus, Ohio.

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.

Christopher D. Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press; 1st edition.

I Dan Melamed. 1996. Automatic construction of clean broad coverage translation lexicons. In *Proceedings of the 2nd Conference of the Association for Machine Translation in the Americas (AMTA'96)*, Montreal, Canada.

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.

P. Proctor, 1988. *Longman English-Chinese Dictionary of Contemporary English*, Longman Group (Far East) Ltd., Hong Kong.

Sinorama. 2002. *Sinorama Magazine.* http://www.greatman.com.tw/sinorama.htm.

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*.

Frank Z. Smadja, Kathleen McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: a statistical approach. *Computational Linguistics,* 22(1):1-38.

Keita Tsuji. 2002. Automatic extraction of translational Japanese-KATAKANA and English word pairs from bilingual corpora. *International Journal of Computer Processing of Oriental Languages*, 15(3):261-279.

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.

Dekai Wu and Xuanyin Xia. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, pages 206–213.