Word-Transliteration Alignment

Tracy Lin
Dep. of Communication Engineering
National Chiao Tung University,
1001, Ta Hsueh Road,
Hsinchu, 300, Taiwan
tracylin@cm.nctu.edu.tw

Chien-Cheng Wu
Department of Computer Science
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
g904374@oz.nthu.edu.tw

Jason S. Chang
Department of Computer Science
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
jschang@cs.nthu.edu.tw

Abstract

The named-entity phrases in free text represent a formidable challenge to text analysis. Translating a named-entity is important for the task of Cross Language Information Retrieval and Question Answering. However, both tasks are not easy to handle because named-entities found in free text are often not listed in a monolingual or bilingual dictionary. Although it is possible to identify and translate named-entities on the fly without a list of proper names and transliterations, an extensive list certainly will ensure the high accuracy rate of text analysis. We use a list of proper names and transliterations to train a *Machine Transliteration Model*. With the model it is possible to extract proper names and their transliterations in a bilingual corpus with high average precision and recall rates.

1. Introduction

Multilingual named entity identification and (back) transliteration has been increasingly recognized as an important research area for many applications, including machine translation (MT), cross language information retrieval (CLIR), and question answering (QA). These transliterated words are often domain-specific and 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.

Transliteration of proper names tends to be varied from translator to translator. Consensus on transliteration of celebrated place and person names emerges over a short period of inconsistency and stays unique and unchanged thereafter. But for less known persons and unfamiliar places, the transliterations of names may vary a great deal. That is exacerbated by different systems used for Ramanizing Chinese or Japanese person and place names. For back transliteration task of converting many transliterations back to the unique original name, there is one and only solution. So back transliteration is considered more difficult than transliteration. Knight and Graehl (1998) pioneered the study of machine transliteration and proposed a statistical transliteration model from English to Japanese to experiment on back transliteration of Japanese named entities. Most previous approaches to machine transliteration (Al-Onaizan and Knight, 2002; Chen et al., 1998; Lin and Chen, 2002); English/Japanese (Knight and Graehl, 1998; Lee and Choi, 1997; Oh and Choi, 2002) focused on the tasks of transliteration and back-transliteration. Very little has been touched upon for the issue of aligning and acquiring words and transliterations in a parallel corpus.

The alternative to on-the-fly (back) machine transliteration is simple lookup in an extensive list automatically acquired from parallel corpora. Most instances of (back) transliteration of proper names can often be found in a parallel corpus of substantial size and relevant to the task. For instance, fifty topics of the CLIR task in the NTCIR 3 evaluation conference contain many named entities (NEs) that require (back) transliteration. The CLIR task involves document retrieval from a collection of late 1990s news articles published in Taiwan. Most of those NEs and transliterations can be found in the articles from the Sinorama Corpus of parallel Chinese-English articles dated from 1990 to 2001, including "Bill Clinton," "Chernobyl," "Chiayi," "Han dynasty," "James Soong," "Kosovo," "Mount Ali," "Nobel Prize," "Oscar," "Titanic," and "Zhu Rong Ji." Therefore it is important for CLIR research that we align and extract words and transliterations in a parallel corpus.

In this paper, we propose a new machine transliteration method based on a statistical model trained automatically on a bilingual proper name list via unsupervised learning. We also describe how the parameters in the model can be estimated and smoothed for best results. Moreover, we show how the model can be applied to align and extract words and their transliterations in a parallel corpus.

The remainder of the paper is organized as follows: Section 2 lays out the model and describes how to apply the model to align word and transliteration. Section 3 describes how the model is trained on a set of proper names and transliterations. Section 4 describes experiments and evaluation. Section 5 contains discussion and we conclude in Section 6.

2. Machine Transliteration Model

We will first illustrate our approach with examples. A formal treatment of the approach will follow in Section 2.2.

2.1 Examples

Consider the case where one is to convert a word in English into another language, says Chinese, based on its phonemes rather than meaning. For instance, consider transliteration of the word “Stanford,” into Chinese. The most common transliteration of “Stanford” is “史丹福.” (Ramanization: [shi-dan-fo]). We assume that transliteration is a piecemeal, statistical process, converting one to six letters at a time to a Chinese character. For instance, to transliterate “Stanford,” the word is broken into “s,” “tan,” “for,” and “d,” which are converted into zero to two Chinese characters independently. Those fragments of the word in question are called transliteration units (TUs). In this case, the TU “s” is converted to the Chinese character “史,” “tan” to “丹,” “for” to “佛,” and “d” to the empty string λ. In other words, we model the transliteration process based on independence of conversion of TUs. Therefore, we have the *transliteration probability* of getting the transliteration “史丹福” given “Stanford,” P(史丹佛 | Stanford),

P(史丹佛 | Stanford) = P(史 | s) P(丹 | tan) P(佛 | for) P(λ | d)

There are several ways such a machine transliteration model (MTM) can be applied, including (1) *transliteration* of proper names (2) *back transliteration* to the original proper name (3) *word-transliteration alignment* in a parallel corpus. We formulate those three problems based on the probabilistic function under MTM:

Transliteration problem (TP)

Given a word *w* (usually a proper noun) in a language (L1), produce automatically the transliteration *t* in another language (L2). For instance, the transliterations in (2) are the results of solving the TP for four given words in (1).

(1) Berg, Stanford, Nobel, 清華
(2) 伯格, 史丹佛, 諾貝爾, Tsing Hua

Back transliteration Problem (BTP)

Given a transliteration *t* in a language (L2), produce automatically the original word *w* in (L1) that gives rise to *t*. For instance, the words in (4) are the results of solving the BTP for two given transliterations in (3).

(3) 米開朗基羅, Lin Ku-fang
(4) Michelangelo, 林谷芳

Word Transliteration Alignment Problem (WTAP)

Given a pair of sentence and translation counterpart, align the words and transliterations therein. For instance, given (5a) and (5b), the alignment results are the three word-transliteration pairs in (6), while the two pairs of word and back transliteration in (8) are the results of solving WTAP for (7a) and (7b)

(5a) Paul Berg, professor emeritus of biology at Stanford University and a Nobel laureate, …
(5b) 史丹佛大學生物系的榮譽教授，諾貝爾獎得主伯格[1]，
(6) (Stanford, 史丹福), (Nobel, 諾貝爾), (Berg, 伯格)
(7a) PRC premier Zhu Rongji's saber-rattling speech on the eve of the election is also seen as having aroused resentment among Taiwan's electorate, and thus given Chen Shui-bian a last-minute boost.
(7b) 而中共總理朱鎔基選前威脅台灣選民的談話，也被認為是造成選民反感，轉而支持陳水扁的臨門一腳。[2]
(8) (Zhu Rongji, 朱鎔基), (Chen Shui-bian, 陳水扁)

Both transliteration and back transliteration are important for machine translation and cross language information retrieval. For instance, the person and place names are likely not listed in a dictionary, therefore should be mapped to the target language via run-time transliteration. Similarly, a large percentage of

[1] Scientific American, US and Taiwan editions. What Clones? Were claims of the first human embryo premature? Gary Stix and 潘震澤(Trans.) December 24, 2001.

keywords in a cross language query are person and place names. It is important for an information system to produce appropriate counterpart names in the language of documents being searched. Those counterparts can be obtained via direct transliteration based on the machine transliteration and language models (of proper names in the target language).

The memory-based alternative is to find those word-transliteration in the aligned sentences in a parallel corpus (Chuang, You, and Chang 2002). Word-transliteration alignment problem certainly can be dealt with based on lexical statistics (Gale and Church 1992; Melamed 2000). However, lexical statistics is known to be very ineffective for low-frequency words (Dunning 1993). We propose to attack WTAP at the sub-lexical, phoneme level.

2.2 The Model

We propose a new way for modeling transliteration of an English word *w* into Chinese *t* via a Machine Transliteration Model. We assume that transliteration is carried out by decomposing *w* into *k* translation units (TUs), $\omega_1, \omega_2, \ldots, \omega_k$ which are subsequently converted independently into $\tau_1, \tau_2, \ldots, \tau_k$ respectively. Finally, $\tau_1, \tau_2, \ldots, \tau_k$ are put together, forming *t* as output. Therefore, the probability of converting *w* into *t* can be expressed as P(*t* | *w*) = $\max_{k, \omega_1 \ldots \omega_k, \tau_1 \ldots \tau_k} \prod_{i=1,k} P(\tau_i \mid \omega_i)$, where $w = \omega_1 \omega_2 \ldots \omega_k$, $t = \tau_1 \tau_2 \ldots \tau_k$, $|t| \le k \le |t|+|w|$, $\tau_i \, \omega_i \neq \lambda$. See Equation (1) in Figure 1 for more details.

Based on MTM, we can formulate the solution to the Transliteration Problem by optimizing P(*t* | *w*) for the given *w*. On the other hand, we can formulate the solution to the Back Transliteration Problem by optimizing P(*t* | *w*) P(*w*) for the given *t*. See Equations (2) through (4) in Figure 1 for more details.

[2] Sinorama Chinese-English Magazine, A New Leader for the New Century--Chen Elected President, April 2000, p. 13.

The word-transliteration alignment process may be handled by first finding the proper names in English and matching up with the transliteration for each proper name. For instance, consider the following sentences in the Sinorama Corpus:

(9c) 「當你完全了解了太陽、大氣層以及地球的運轉，你仍會錯過了落日的霞輝，」西洋哲學家懷海德說。

(9e) "When you understand all about the sun and all about the atmosphere and all about the rotation of the earth, you may still miss the radiance of the sunset." So wrote English philosopher Alfred North Whitehead.

It is not difficult to build part of speech tagger or named entity recognizer for finding the following proper names (PN):

(10a) Alfred, (10b) North, (10c) Whitehead.

We use Equation (5) in Figure 1 to model the alignment of a word w and its transliteration t in s based on the *alignment probability* P(s, w) which is the product of transliteration probability P($\sigma|\ \omega$) and a trigram match probability, P($m_{\mathrm{i}}\ |\ m_{\mathrm{i-2}}$, $m_{\mathrm{i-1}}$), where m_{i} is the type of the i-th match in the alignment path. We define three match types based on lengths a and b, $a = |\ \tau\,|$, $b = |\ \omega\,|$: match(a, b) = H if a = 0, match(a, b) = V if b = 0, and match(a, b) = D if a > 0 and b > 0. The D-match represents a non-empty TU ω matching a transliteration character τ, while the V-match represents the English letters omitted in the transliteration process.

MACHINE TRANSLITERATION MODEL: The probability of transliteration t of the word w

$$\mathrm{P}(t \mid w) = \max_{k,\omega_1 \ldots \omega_k, \tau_1 \ldots \tau_k} \prod_{i=1,k} P(\tau_i \mid \omega_i), \tag{1}$$

where $w = \omega_1 \omega_2 \ldots \omega_k$,
$t = \tau_1 \tau_2 \ldots \tau_k$,
$| t | \leq k \leq | t | + | w |$,
$| \tau_i \, \omega_i | \geq 1$.

TRANSLITERATION: Produce the phonetic translation equivalent t for the given word w

$$t = \arg\max_{t} \mathrm{P}(t \mid w) \tag{2}$$

BACK TRANSLITERATION: Produce the original word w for the given transliteration t

$$\mathrm{P}(w \mid t) = \frac{\mathrm{P}(t \mid w)\,\mathrm{P}(w)}{\mathrm{P}(t)} \tag{3}$$

$$w = \arg\max_{t} \frac{\mathrm{P}(t \mid w)\,\mathrm{P}(w)}{\mathrm{P}(t)} = \arg\max_{t} \mathrm{P}(t \mid w)\,\mathrm{P}(w) \tag{4}$$

WORD-TRANSLITERATION ALIGNMENT: Align a word w with its transliteration t in a sentence s

$$\mathrm{P}(s, w) = \max_{k,\omega_1 \ldots \omega_k, \sigma_1 \ldots \sigma_k} \prod_{i=1,k} \mathrm{P}(\sigma_i \mid \omega_i)\,\mathrm{P}(m_i \mid m_{i-2}, m_{i-1}), \tag{5}$$

where $w = \omega_1 \omega_2 \ldots \omega_K$,
$s = \sigma_1 \sigma_2 \ldots \sigma_K$, (both ω_i and σ_i can be empty)
$| s | \leq k \leq | w | + | s |$, $|\omega_i \, \sigma_i| \geq 1$,
m_i is the type of the (ω_i , σ_i) match, $m_i = \mathrm{match}(|\omega_i|, |\sigma_i|)$,
$\mathrm{match}(a, b) = H$, if $b = 0$,
$\mathrm{match}(a, b) = V$, if $a = 0$,
$\mathrm{match}(a, b) = D$, if $a > 0$ and $b > 0$,
$\mathrm{P}(m_i \mid m_{i-2}, m_{i-1})$ is trigram Markov model probabiltiy of match types.

$$\alpha(i, j) = \mathrm{P}(s_{1:i-1}, w_{1:j-1}). \tag{6}$$

$$\alpha(1, 1) = 1, \ \mu(1, 1) = (H, H). \tag{7}$$

$$\alpha(i, j) = \max_{a=0,1,\, b=0,6} \alpha(i\text{-}a, j\text{-}b)\,\mathrm{P}(s_{j-a:j-1} \mid w_{i-b:i-1})\,\mathrm{P}(\,\mathrm{match}(a, b) \mid \mu(i\text{-}a, j\text{-}b)\,). \tag{8}$$

$$\mu(i, j) = (m, \mathrm{match}(a^*, b^*)), \text{ where } \mu(i\text{-}a^*, j\text{-}b^*) = (x, m), \tag{9}$$

where $(a^*, b^*) = \arg\max_{a=0,1,\, b=0,6} \alpha(i\text{-}a, j\text{-}b)\,\mathrm{P}(s_{j-a:j-1} \mid w_{i-b:i-1})\,\mathrm{P}(\,\mathrm{match}(a, b) \mid \mu(i\text{-}a, j\text{-}b)\,)$.

Figure 1. The equations for finding the Viterbi path of matching a proper name and its translation in a sentence

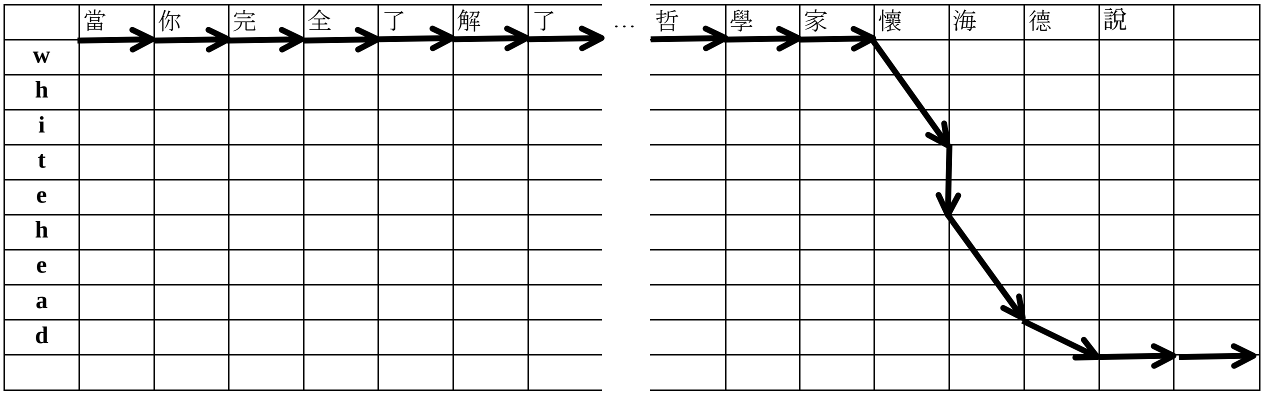

Figure 2. The Viterbi alignment path for Example (9c) and the proper name "Whitehead" (10c) in the sentence (9e), consisting of one V-match (te-λ), three D-matches (whi–懷, hea–海, d–德), and many H-matches.

To compute the alignment probability efficiently, we need to define and calculate the *forward probability* $\alpha(i, j)$ of $P(s, w)$ via dynamic programming (Manning and Schutze 1999), $\alpha(i, j)$ denotes the probability of aligning the first i Chinese characters of s and the first j English letters of w. For the match type trigram in Equation (5) and (8), we need also compute $\mu(i, j)$, the types of the last two matches in the Viterbi alignment path. See Equations (5) through (9) in Figure 1 for more details.

For instance, given w = "Whitehead" and s = "「當你完全了解了太陽、大氣層以及地球的運轉,你仍會錯過了落日的霞輝，」西洋哲學家懷海德說。," the best Viterbi path indicates a decomposition of word "Whitehead" into four TUs, "whi," "te," "hea," and "d" matching "懷," λ, "海," "德" respectively. By extracting the sequence of *D*- and *V*-matches, we generate the result of word-transliteration alignment. For instance, we will have (懷海德, Whitehead) as the output. See Figure 2 for more details.

3. Estimation of Model Parameters

In the training phase, we estimate the transliteration probability function $P(\tau|\ \omega)$, for any given TU ω and transliteration character τ, based on a given list of word-transliterations. Based on the Expectation Maximization (EM) algorithm (Dempster et al., 1977) with Viterbi decoding (Forney, 1973), the iterative parameter estimation procedure on a training data of word-transliteration list, (E_k, C_k), $k = 1$ to n is described as follows:

Initialization Step:

Initially, we have a simple model $P_0(\tau|\ \omega)$

$$\begin{aligned} P_0(\tau|\ \omega) &= \mathrm{sim}(\ R(\tau)\ |\ \omega) \\ &= \mathrm{dice}(t_1 t_2 \ldots t_a, w_1 w_2 \ldots w_b) \qquad (8) \\ &= \frac{2c}{a+b} \end{aligned}$$

where $R(\tau)$ = Romanization of Chinese character τ

$R(\tau) = t_1 t_2 \ldots t_a$

$\omega = w_1 w_2 \ldots w_b$

c = # of common letters between $R(\tau)$ and ω

For instance, given w = ‘*Nayyar*’ and t = ‘納雅,’ we have and R(τ_1) = ‘na’ and R(τ_2) = ‘ya’ under Yanyu Pinyin Romanization System. Therefore, breaking up w into two TUs, ω_1 = ‘nay’ ω_2 = ‘yar’ is most probable, since that maximizes $P_0(\tau_1 \mid \omega_1) \times P_0(\tau_2 \mid \omega_2)$

$P_0(\tau_1 \mid \omega_1)$= sim(na | *nay*) = 2 × 2 / (2+3) = 0.8
$P_0(\tau_2 \mid \omega_2)$= sim(ya | *yar*) = 2 × 2 / (2+3) = 0.8

Expectation Step:

In the Expectation Step, we find the best way to describe how a word get transliterated via decomposition into TUs which amounts to finding the best Viterbi path aligning TUs in E_k and characters in C_k for all pairs (E_k, C_k), k = 1 to n, in the training set. This can be done using Equations (5) through (9). In the training phase, we have slightly different situation of $s = t$.

Table 1. The results of using $P_0(\tau|\omega)$ to align TUs and transliteration characters

w	s=t	ω-τ match on Viterbi path
Spagna	斯帕尼亞	s-斯 pag-帕 n-尼 a-亞
Kohn	孔恩	koh-孔 n-恩
Nayyar	納雅	nay-納 yar-雅
Alivisatos	阿利維撒托斯	a-阿 li-利 vi-維 sa-撒 to-托 s-斯
Rivard	里瓦德	ri-里 var-瓦 d-德
Hall	霍爾	ha-霍 ll-爾
Kalam	卡藍	ka-卡 lam-藍
Salam	薩萊姆	sa-薩 la-萊 m-姆
Adam	亞當	a-亞 dam-當
Gamoran	蓋莫藍	ga-蓋 mo-莫 ran-藍
Heller	赫勒	hel-赫 ler-勒
Adelaide	阿得雷德	a-阿 de-得 lai-雷 de-德
Nusser	努瑟	nu-努 sser-瑟
Nechayev	納卡耶夫	ne-納 cha-卡 ye-耶 v-夫
Hitler	希特勒	hi-希 t-特 ler-勒
Hunt	杭特	hun-杭 t-特
Germain	杰曼	ger-杰 main-曼
Massoud	馬蘇德	ma-馬 ssou-蘇 d-德
Malong	瑪隆	ma-瑪 long-隆
Gore	高爾	go-高 re-爾
Teich	泰許	tei-泰 ch-許
Laxson	拉克森	la-拉 x-克 son-森

The Viterbi path can be found via a dynamic programming process of calculating the forward probability function α(i, j) of the transliteration alignment probability P(E_k, C_k) for $0 < i < | C_k |$ and $0 < j < | E_k |$. After calculating P(C_k, E_k) via dynamic programming, we also obtain the TU matches (τ, ω) on the

Viterbi path. After all pairs are processed and TUs and translation characters are found, we then re-estimate the transliteration probability P(τ | ω) in the Maximization Step

Maximization Step:
Based on all the TU alignment pairs obtained in the Expectation Step, we update the maximum likelihood estimates (MLE) of model parameters using Equation (9).

$$\mathrm{P}_{MLE}(\tau \mid \omega) = \frac{\sum_{i=1}^{n} \sum_{\tau \text{ matches } \omega \text{ in } (E_{\mathrm{i}}, C_{\mathrm{i}})} \mathrm{count}(\tau, \omega)}{\sum_{i=1}^{n} \sum_{\tau' \text{ matches } \omega \text{ in } (E_{\mathrm{i}}, C_{\mathrm{i}})} \mathrm{count}(\omega)} \qquad (9)$$

The Viterbi EM algorithm iterates between the Expectation Step and Maximization Step, until a stopping criterion is reached or after a predefined number of iterations. Re-estimation of P(τ | ω) leads to convergence under the Viterbi EM algorithm.

3.1 Parameter Smoothing

The maximum likelihood estimate is generally *not* suitable for statistical inference of parameters in the proposed machine transliteration model due to data sparseness (even if we use a longer list of names for training, the problem still exists). MLE is not capturing the fact that there are other transliteration possibilities that we may have not encountered. For instance, consider the task of aligning the word “Michelangelo” and the transliteration “米開朗基羅” in Example (11):

(11) (Michelangelo, 米開朗基羅)

It turns out in the model trained on some word-transliteration data provides the MLE parameters in the MTM in Table 2. Understandably, the MLE-based model assigns 0 probability to a lot of cases not seen in the training data and that could lead to problems in word-transliteration alignment. For instance, relevant parameters for Example (11) such as P(開 | che) and P(朗 | lan) are given 0 probability. Good Turing estimation is one of the most commonly used approaches to deal with the problems caused by data sparseness and zero probability. However, GTE assigns identical probabilistic values to all unseen events, which might lead to problem in our case.

Table 2. $P_{MLE}(t \mid n)$ value relevant to Example (11)

English TU ω	**Transliteration τ**	$P_{MLE}(\tau \mid \omega)$
mi	米	**0.00394**
mi	密	0.00360
mi	明	0.00034
mi	麥	0.00034
mi	邁	0.00017
che	傑	0.00034
che	切	0.00017
che	其	0.00017
che	奇	0.00017
che	契	0.00017
che	科	0.00017
che	**開**	**0**
lan	蘭	0.00394
lan	藍	0.00051
lan	倫	0.00017
lan	**朗**	**0**
ge	格	0.00102
ge	奇	0.00085
ge	吉	0.00068
ge	**基**	**0.00017**
ge	蓋	0.00017
lo	洛	0.00342
lo	**羅**	**0.00171**
lo	拉	0.00017

We observed that although there is great variation in Chinese transliteration characters for any given English word, the initial, mostly consonants, tend to be consistent. See Table 3 for more details. Based on that observation, we use the linear interpolation of the Good-Turing estimation of TU-to-TU and the class-based initial-to-initial function to approximate the parameters in MTM. Therefore, we have

$$P_{li}(c \mid e) = 0.5\, P_{GT}(c \mid e) + 0.5\, P_{MLE}(\text{init}(c) \mid \text{init}(e))$$

4 Experiments and evaluation

We have carried out rigorous evaluation on an implementation of the method proposed in this paper. Close examination of the experimental results reveal that the machine transliteration is general effective in aligning and extracting proper names and their transliterations from a parallel corpus.

The parameters of the transliteration model were trained on some 1,700 proper names and transliterations from Scientific American Magazine. We place 10 *H*-matches before and after the Viterbi alignment path to simulate the word-transliteration situation and trained the trigram match type probability. Table 4 shows the estimates of the trigram model.

Table 3. The initial to initial correpsondence of ω amd R(τ)

ω	**τ**	**R(τ)**	**Init(ω)**	**Init(R(τ))**
mi	米	mi	m	m
mi	密	mi	m	m
mi	明	min	m	m
mi	麥	mai	m	m
mi	邁	mai	m	m
che	傑	jei	ch	j
che	切	chei	ch	ch
che	其	chi	ch	ch
che	奇	chi	ch	ch
che	契	chi	ch	ch
che	科	ke	ch	k
che	開	kai	ch	k
lan	蘭	lan	l	l
lan	藍	lan	l	l
lan	倫	lun	l	l
lan	朗	lang	l	l
ge	格	ge	g	g
ge	奇	chi	g	ch
ge	吉	ji	g	j
ge	基	ji	g	j
ge	蓋	gai	g	g
lo	洛	lo	l	l
lo	羅	Lo	l	l
lo	拉	La	l	l

Table 4. The stastical estimates of trigram match types

Match Type Trigram $m_1 m_2 m_3$	Count	$P(m_3 \mid m_1 m_2)$
DDD	1886	0.51
DDH	1627	0.44
DDV	174	0.05
DHD	0	0.00
DHH	1702	1.00
DHV	0	0.00
DVD	115	0.48
DVH	113	0.47
DVV	12	0.05
HDD	1742	0.96
HDH	7	0.01
HDV	58	0.03
HHD	1807	0.06
HHH	29152	0.94
HHV	15	0.00
HVD	15	1.00
HVH	0	0.00

The model was then tested on three sets of test data:

(1) 200 bilingual examples in Longman Dictionary of Comtemporary Dictionary, English-Chinese Edition.
(2) 200 aligned sentences from Scientific American, US and Taiwan Editions.
(3) 200 aligned sentences from the Sinorama Corpus.

Table 5 shows that on the average the precision rate of exact match is between 75-90%, while the precision rate for character level partial match is from 90-95%. The average recall rates are about the same as the precision rates.

Table 5. The experimental results of word-transliteration alignement

Test Data	# of words (# of characters)	# of matches (# of characters)	Word precision (Characters)
LODCE	200	179	89.5%
	(496)	(470)	(94.8%)
Sinorama	200	151	75.5%
	(512)	(457)	(89.3%)
Sci. Am.	200	180	90.0%
	(602)	(580)	(96.3%)

5. Discussion

The success of the proposed method for the most part has to do with the capability to balance the conflicting needs of capturing lexical preference of transliteration and smoothing to cope with data sparseness and generality. Although we experimented with a model trained on English to Chinese transliteration, the model seemed to perform reasonably well even with situations in the opposite direction, Chinese to English transliteration. This indicates that the model with the parameter estimation method is very general in terms of dealing with unseen events and bi-directionality.

We have restricted our discussion and experiments to transliteration of proper names. While it is commonplace for Japanese to have transliteration of common nouns, transliteration of Chinese common nouns into English is rare. It seems that is so only when the term is culture-specific and there is no counterparts in the West. For instance, most instances "旗袍" and "瘦金體" found in the Sinorama corpus are mapped into lower case transliterations as shown in Example (11) and (12):

(11a) 中國國服——旗袍真的沒落了嗎？
(11b) Are ch'i-p'aos--the national dress of China--really out of fashion?

(12a) 一幅瘦金體書法複製品
(12b) a scroll of shou chin ti calligraphy

Without capitalized transliterations, it remains to be seen how word-transliteration alignment related to common nouns should be handled.

6. Conclusion

In this paper, we propose a new statistical machine transliteration model and describe how to apply the model to extract words and transliterations in a parallel corpus. The model was first trained on a modest list of names and transliteration. The training resulted in a set of ‘syllabus’ to character transliteration probabilities, which are subsequently used to extract proper names and transliterations in a parallel corpus. These named entities are crucial for the development of named entity identification module in CLIR and QA.

We carried out experiments on an implementation of the word-transliteration alignment algorithms and tested on three sets of test data. The evaluation showed that very high precision rates were achieved.

A number of interesting future directions present themselves. First, it would be interesting to see how effectively we can port and apply the method to other language pairs such as English-Japanese and English-Korean. We are also investigating the advantages of incorporate a machine transliteration module in sentence and word alignment of parallel corpora.

Acknowledgement

We acknowledge the support for this study through grants from National Science Council and Ministry of Education, Taiwan (NSC 90-2411-H-007-033-MC and MOE EX-91-E-FA06-4-4).

References

Al-Onaizan, Y. and K. 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.

Chen, H.H., S-J Huang, Y-W Ding, and S-C Tsai. 1998. Proper name translation in cross-language information retrieval. In *Proceedings of 17th COLING and 36th ACL*, pages 232-236.

Chuang, T., G.N. You, J.S. Chang (2002) Adaptive Bilingual Sentence Alignment, *Lecture Notes in Artificial Intelligence 2499*, 21-30.

Cibelli, J.B. R.P. Lanza, M.D. West, and C. Ezzell. 2002. What Clones? SCIENTIFIC AMERICAN, Inc., New York, January. http://www.sciam.com.

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.

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

Dempster, A.P., 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.

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

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

Lee, J.S. and K-S 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.

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

Manning, Ch. and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press; 1st edition.

Oh, J-H and K-S 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.

Proctor, P. 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.

Stalls, B.G. and K. Knight. 1998. Translating names and technical terms in Arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

Tsujii, K. 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.