Exploiting Association Words to Retrieve Synonymous Transliterations from Web Snippets

Chien-Hsing Chen
Department of Information Management,
National Yunlin University of Science and Technology

Chung-Chian Hsu†
Department of Information Management,
National Yunlin University of Science and Technology

Abstract

There is no translation standard across the regions such as Taiwan, Hong Kong and China where Chinese language is used. As a result, a foreign proper noun is often translated to different Chinese words which lead to the incomplete search problem if only one of the words is used as the query keyword to a search engine. In this paper, we present a framework to retrieve synonymous transliterations as many as possible from the Web for an input Chinese transliteration. The research results could be applied to query expansion so as to alleviate the incomplete search problem. There are two major phases in the framework. The first is to develop an effective method to collect relevant Web snippets which may contain synonymous transliterations. The second is to extract synonymous transliterations from the set of relevant Web snippets. Experimental results show that the proposed framework is feasible and effective. Moreover, most of extracted synonymous transliterations, compared with other noise terms, have a higher rank of similarity to the input transliteration.

Keywords: text mining, Web mining, synonymous transliteration, cross lingual information retrieval, Chinese transliteration

† The Corresponding Author

利用關聯詞從全球資訊網中探勘同義音譯詞

陳建興
國立雲林科技大學資訊管理學系

許中川*
國立雲林科技大學資訊管理學系

摘要

使用中文語系的地方如台灣、香港與中國，並沒有統一的翻譯標準，以致於同一個外來詞通常被翻譯成數個不同的中文詞。例如，澳洲首都Sydney依其發音被翻譯成「雪梨」、「雪黎」或「悉尼」等不同的中文音譯詞。如此的翻譯結果，會導致搜尋引擎檢索資料不完整。例如，使用「雪梨」檢索，無法得到使用「雪黎」與「悉尼」翻譯詞的網頁資料。本研究我們提出一套探勘架構：給予一個中文音譯詞，透過搜尋全球資訊網網頁，盡可能找出其所有的中文同義音譯詞。本研究成果可應用於改善搜索引擎跨語系資料檢索不齊全之問題。研究架構包括兩個階段，首先，我們提出一個有效率的方法蒐集有可能包含同義音譯詞的相關網頁摘要短文。其次是從蒐集的網頁摘要短文中萃取同義的音譯詞。實驗結果證明我們所提方法的可行性，顯示可以有效地找到許多同義音譯詞。再者，找到的同義音譯詞和其他雜訊相比，大部分對輸入音譯詞都有比較高的相似度排名。

關鍵字： 文字探勘, 網頁探勘, 同義音譯詞, 跨語言資訊檢索, 中文音譯詞

* 通訊作者

1. INTRODUCTION

Transliteration is a representation of a foreign proper noun by rendering its pronunciation to a local language. With many different translators working without a common standard, there may be several Chinese transliterations for the same proper noun, especially for personal names and geographical names. For example, inconsistent Chinese transliterations, 戈巴契夫 (ge ba qi fu), 哥巴卓夫 (ge ba zhuo fu) and 戈爾巴喬夫 (ge er ba qiao fu), are all transliterated from a cognate name "Gorbachev". An Australia city "Sydney" has different transliterations including 雪梨 (xue li), 雪黎 (xue li) and 悉尼 (xi ni). Someone with the Chinese language as his native language would never know all the Chinese synonymous transliterations of a foreign word. Indeed, the situation of Chinese transliteration variation leads to reading confusion and moreover incomplete search results of Web pages. When one submits one of the transliterations, say 雪梨, as the search-keyword to a search engine, one gets only the pages of 雪梨 but no pages of 悉尼. Chinese is used in many regions such as Taiwan, Hong Kong and China and constitutes a large portion of users. The incomplete search problem will result in missing critical information during collecting data from the Web. In this research, we attempt to retrieve synonymous transliterations as many as possible from the Web for a given Chinese transliteration. The research result can be applied to construct a database of synonymous transliterations for automatic query expansion so as to help reduce the impact of the incomplete search problem.

Some major tasks in natural language processing such as machine translation (MT), named entity recognition (NER), information extraction (IE) and cross-language information retrieval (CLIR) have treated the Web as a huge corpus for extracting knowledge or valuable information. It is worth noting that extracting transliterations from the Web corpus (Kuo et al. 2007) instead of comparing collected parallel datasets (Brill et al. 2001; Collier & Hirakawa 1997; Hsu et al. 2007; Lin & Chen 2002; Tsuji 2002; Virga & Khudanpur 2003) is straightforward; thus collecting an appropriate subset from the Web corpus is one of the most important processes. The techniques in collecting an appropriate set include content focused crawler (Aggarwal et al. 2001; Babaria et al. 2007; Barbosa & Freire 2007) and meta-search engine (Oztekin et al. 2002; Qin et al. 2004; Selberg & Etzioni 1997). However, they are usually faced with determining a seed (root) Web site, leading to a time-consuming task in visiting a large number of Web pages.

Search engines have been considered as an important knack to retrieve the documents which contain the search keyword(s) interesting to the user. Since we want to collect as many synonymous transliterations as possible from the Web, using appropriate search keywords to retrieve relevant Web snippets which may contain target transliterations is a key step. However, a simple query without any spice may fail in returning useful documents because a short or

inadequate query may retrieve trivial Web pages (Oyama et al. 2004). Therefore, a search keyword should be made more delicate so as to aid in retrieving highly relevant Web documents. Such the paradigm as reported in the tasks of query extension (Carpineto et al. 2001) and keyword spice (Oyama et al. 2004) attempted to determine the spiced search-keywords which are determined from a knowledge corpus. Nevertheless, it is not scalable and brings a great cost of designing and manipulating an effective corpus for obtaining appropriate search keywords.

In this paper we present a framework consisting of two major procedures. Instead of visiting and downloading a huge amount of documents from the Web, and requiring of constructing an effective corpus for learning the search keywords, our first procedure is to develop an effective method to collect relevant Web snippets by the use of a search engine. The search keywords are determined by an integrated statistical approach which uses the returned page count from the search engine and never requires processing the corpus in order to obtain the page count.

The second procedure is to extract Chinese transliteration candidates from the free-text snippets by the help of information retrieval techniques, and then we construct a decision model to identify synonymous transliterations from the set of the candidate terms. The decision model consists of two major components. First, we employ a digitized sound comparison technique to measure pronunciation similarity between the transliteration and its candidates. Second, we derive a context comparison approach to measuring their semantic similarity. The context of a transliteration with its candidates is built on the returned Web snippets of a search engine. Finally, the similarity between the input transliteration and the extracted candidates is calculated by combination of the pronunciation and the semantic similarity.

We attempt to tackle the issue of mining as many synonymous transliterations as possible from the Web with respect to a given transliteration. Unlike many studies in IR or CLIR, our framework does not need a pre-collected training corpus, which suffers from bias if the collection is not comprehensive, nor involve manual assignment between phonemes, which could be subjective and tedious. Moreover, the proposed approach involves processing only Web snippets rather than whole Web pages so that the computation load is less demanding.

The rest of the article is organized as follows. Related work is described in Section 2. Section 3 presents a procedure and describes the details of collecting the potential Web snippets in which synonymous transliterations possibly appear. The process of extracting synonymous transliterations from the collected Web snippets is presented in Section 4. Experimental results are given in Section 5. Finally, Section 6 gives the conclusions.

2. RELATED WORK

A transliteration represents a foreign word in a local language by rendering the

pronunciation of the foreign word in the alphabet to the local language, such as “Gorbachev” transliterated by “戈巴契夫” in Chinese. The issue of handling proper noun transliterations, in particular, identifying pairs of corresponding proper nouns from bilingual corpora has long been studied in (Brill et al. 2001; Collier & Hirakawa 1997; Lin & Chen 2002; Tsuji 2002; Virga & Khudanpur 2003). The problem can be classified into two directions includes forward transliteration and backward transliteration (Knight & Graehl 1998). Forward transliteration is the process of phonetically convert an original proper noun in the source language to a transliterated word in the target language, i.e., from “Gorbachev” to “戈巴契夫”. Backward transliteration works oppositely from a transliterated word to its original source word, i.e., from “戈巴契夫” to “Gorbachev”(Chen et al. 1998; Chen et al. 2006; Lin & Chen 2000; Lin & Chen 2002; Stalls & Knight 1998).

2.1 Extracting Transliteration from the Web

Recently, a lot of research has utilized abundant Web resources for various issues of cross-lingual information retrieval, includes transliteration extraction. Lu et al. proposed approaches to generate translation suggestions for given user queries via mining anchor text and search results (Lu et al. 2002; Lu et al. 2003). Li et al. developed an intelligent English reading-assistance system that offers word and phrase translation based on multilingual Web data and statistical-learning methods (Li et al. 2003). Zhang and Vines devised a method to dynamically discover translations of out of vocabulary terms (Zhang & Vines 2004). Fang et al. proposed an approach to mine English translations of Chinese terms. Given a Chinese term, their method collects effective Web pages based on semantic prediction (Fang et al. 2006). Analysis is performed on the Web pages so as to further collect more effective Web pages. Likely English translations in the pages are evaluated according to some predefined features. Wu and Chang presented a method for learning to find English to Chinese transliterations on the Web (Wu & Chang 2007). Sub-lexical relationships between English names and their Chinese transliterations are learned from a set of training data a priori. At run-time, the relationships are used to expand the given English-name query for retrieving Web pages and then are further used to help extract and evaluate candidate Chinese terms. Kuo et al. assume that Chinese transliteration always co-occur in proximity to their original English words and then proposed a phonetic similarity modeling approach to identify the transliteration pairs by measuring phonetic similarity between candidate transliteration pairs (Kuo et al. 2007).

2.2 Similarity Evaluation for Transliterations

The algorithms for calculating the phonetic similarity between two words are relevant to our work, when we need to compare the similarity between an input transliteration and its

extracted candidate. There are several approaches to comparing the similarity of two Chinese words, mainly including physical-sound-, grapheme- and phoneme-based approaches (Hsu et al. 2007). Physical-sound-based approaches measure the similarity of two words based on the similarity between digitalized physical sounds (Hsu et al. 2007). Grapheme-based approaches (Wagner & Fischer 1974) compare the similarity of two strings of Roman alphabets which the two Chinese words are converted to by one of the Pinyin schemes such as Hanyu, Tongyong, Wade-Giles etc. The similarity of two terms is proportional to the size of common alphabets occurred in the two strings. Phoneme-based approaches (Kondrak 2003; Kuo et al. 2007; Lin & Chen 2002) are mainly based on the pronunciation similarity between phones. The approaches take into consideration of articulatory features of phones. For example, in phoneme-based approaches "p" is more similar to "b" than to "m" while in grapheme-based approaches both pairs have the same degree of similarity, i.e., totally dissimilar due to distinct alphabets.

Learning phonetic similarity between alphabets, for instance "b" and "p", has been developed with variant applications in machine transliteration. The phonetic similarity can often be learned from a training corpus (Kuo et al. 2007; Lin & Chen 2002; Wu & Chang 2007) or assigned manually (Lin & Chen 2000).

2.2.1 *Learning Phonetic Similarity from a Training Corpus*

The phonetic similarity model in essence consists of a syllable-based confusion matrix with its element being the conditional probability $p(es|cs)$ where *es* and *cs* are an English syllable and a Chinese syllable, respectively. The probability can be estimated by several proposed methods with the use of various training data including a labeled English speech database or a transliterated bilingual corpus. Usually, it requires an approximate probability distribution referred to a prepared parallel training corpus for learning in rendering the alphabets.

Knight and Graehl proposed a backward phoneme-based transliteration system from Japanese to English with regard to five stages (Knight & Graehl 1998). In their work, they proposed a tree based structure, namely weighted finite-state transducers (WFSTs) organized from a set of English/Japanese sequence pairs based on probabilities and Bayes' theorem, to estimate similarity amongst phonemes. Ai-Onaizan and Knight (Al-Onaizan & Knight 2002) applied the approach proposed in Knight and Graehl (Knight & Graehl 1998) to cover languages between Arabic and English, and compared the performance of the proposed approach to that of human translators. AbdulJaleel and Larkey proposed a statistical grapheme-based transliteration model for converting between the Arabic alphabet and the English alphabet (AbdulJaleel & Larkey 2003). They argued that an Arabic string is generated by a set of individual English characters; thus the Arabic string could be transliterated by an English alphabet with probability. The researches in the literature (Jeong et al. 1999; Meng et al. 2001; Och & Ney 2000) also explored English-Korean transliteration by the use of statistical-based model. Virga and

Khudanpur presented an application of cross-lingual information retrieval based on statistical machine transliteration (Virga & Khudanpur 2003). The system has to compare the similarity score between an English phoneme sequence (generated from an English name) and a Chinese phonetic sequence (generated from a Chinese transliteration via Pinyin Romanization). Li et al. (Li et al. 2004) studied an alignment process to handle the task of English-Chinese transliteration. They argued that the syllables of a Chinese transliteration with corresponding to that of its foreign English term could be aligned via maximum likelihood estimation for the two probability distributions. In (Lin & Chen 2002), a pronouncing dictionary with a modified Widrow-Hoff learning algorithm was used to determine the similarity between 97 phonemes used to represent 1574 training pairs of English and transliterated Chinese names. Lee et al. proposed an approach based on a statistical machine transliteration model to exploit the phonetic similarities between English words and corresponding Chinese transliterations (Lee et al. 2006). Their method does not require a pronouncing dictionary. The parameters of the model are automatically learned from a bilingual proper name list. Oh et al. studied a machine transliteration model based on correspondence between graphemes and phonemes using three learning algorithms including maximum entropy model, decision tree model and memory-based learning model (Oh et al. 2006b). Oh and Choi tested the experiments on English-to-Korean transliteration and English-to-Japanese transliteration, and described a ranking scheme for transliteration extraction from Web data (Oh & Choi 2006a). Gao et al. played a training process of dynamically discovered alignment to map from a set of English phonemes to a set of Romanized Chinese phonetic symbols constituted from a legitimate Chinese character (Tao et al. 2006). They tested the experiments on the English/Arabic, English/Chinese and English/Hindi correspondences by the use of the learned cost matrix. Yoon et al. adopted the phonetic scoring method proposed in (Tao et al. 2006) and trained a linear classifier to achieve a comparable result (Yoon et al. 2007).

2.2.2 *Calculating Phonetic Similarity by Rules*

An alternative approach to measuring phonetic similarity is to use pre-determined similarity rules. The similarity rules are usually constructed by considering the multi-valued phonetic features with respect to the tones and the pronunciation places, such as lip, palate, tongue and bilabial. Two phonetic alphabets having the same pronunciation places indicate that they have high similarity score. For instance, the pronunciation places of both /p/ and /b/ are bilabial and the two are assigned a high similarity score.

In (Lin & Chen 2000), they devise rules to determine the similarity of each pair phones. Connolly proposed a scheme for evaluating the similarity score between phonemes (Connolly 1997). He organized two perceptual features with respect to separating consonant phonemes into six groups. The similarity score of two consonants owning a corresponding group would

be higher than that owning a different one. Chen et al. argued that different phonetic alphabets may own the same sounds and proposed a scheme to compare similarity between Chinese person names and English ones (Chen et al. 1998). Lin and Chen designed a scoring scheme in assigning similarity between phonemes (Lin & Chen 2000). They argued that the similarity score of a matched consonant pair is different than that of a matched vowel pair. They also designed a set of corresponding pairs, such as the pairs {b, p} and {d, t} which are assigned as similar in comparison. Kondrak presented a scoring scheme for computing phonetic similarity between alphabets (Kondrak 2003). He mentioned that equal weight assigned to all features cannot address the problem of unequal relevance of features; thus the scheme considered multi-valued articulatory features and assigned different weights on features.

2.3 Contributions

Different from the previous studies, our research objective is to find out from the Web as many synonymous transliterations as possible for a given Chinese transliteration. We propose a framework and experimentally prove its feasibility. Our method requires neither training data nor manually assigned phonetic similarity scores. Incomprehensive training data would lead to biased estimation (Kondrak 2003). Manual assignment of phonetic similarity is tedious and would be subjective. In addition, our mining framework involves processing only the Web snippets instead of processing the whole Web pages which will be computationally intensive. The framework is effective to mine synonymous transliterations as many as possible by the use of search engine. The research result can be applied to help construct a dataset of synonymous transliterations for alleviating the incomplete search problem.

3. Synonymous Transliterations Extraction from Web Snippets

In this paper we present an approach to mining synonymous transliterations from collected Web snippets by the use of a search engine. Determining an appropriate query strategy is the first step for the collection of relevant Web snippets which may contain synonymous transliterations. The procedure is illustrated in Fig. 1. First, a transliteration (TL) is inputted to collect a set of n snippets, called core snippets. A set of m keywords, called association words, are extracted from the core snippets. We then use different strategies to retrieve a set of Web snippets, called target snippets, which are considered containing synonymous transliterations. We extract Chinese transliteration candidates from the target snippets.

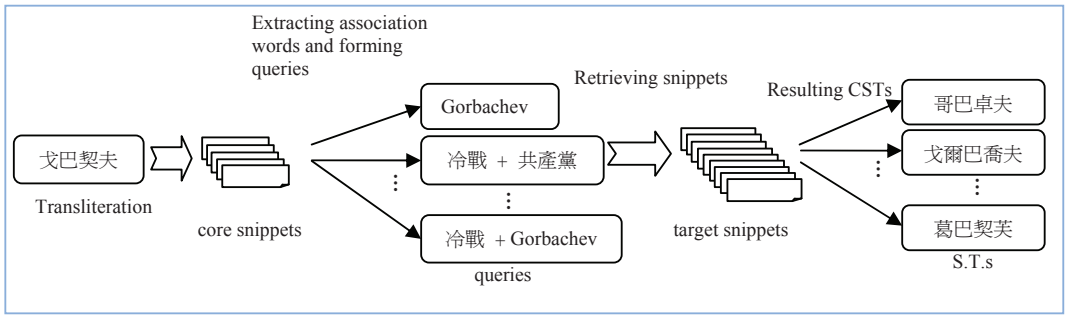

Figure 1: The procedure to retrieve synonymous transliterations from the Web.

3.1 Extract Association words with the Help of a Search Engine

Association words which are highly relevant to the input transliteration have to be identified first. The association strength can be measured via the search engine (Cheng et al. 2004). Cilibrasi and Vitanyi also considered information distance to obtain the similarity among the names of objects by using returning page counts from a search engine (Cilibrasi & Vitanyi 2007). Therefore, we also exploit this idea for our association words determination to avoid using the traditional, complicate techniques in the information retrieval field.

We use an example to depict the process. To determine association strength between 冷戰 and 戈巴契夫 by information gain measure defined in Eq. (1), the probability of *t* and TL co-occurence *p(t, TL)* = *p*(冷戰, 戈巴契夫) is estimated by submitting a query "冷戰" + "戈巴契夫" to the Google search engine and its returning page count is divided by *N*, the number of Web pages, i.e., #{("冷戰" + "戈巴契夫")}/*N*. We used a large number 3 billion for N in our experiment.

$$IG(t,TL) = p(t,TL)\times\log\frac{p(t,TL)}{p(TL)p(t)} + p(\bar{t},TL)\times\log\frac{p(t,TL)}{p(TL)p(\bar{t})} \quad (1)$$

To reduce variation and bias from a single measure (Huang et al. 2006) , we use an integrated statistical measure, which take the average rank of the ranking results from six commonly used measures, including Information gain (IG), Mutual information (MI), Chi-square (CHI), Correlation coefficient (CC), Relevance score (RS), Odds ratio (OR) and GSS Coefficient (GSS).

3.2 Collect Target Snippets

Once association words of an input transliteration are identified, the next step to use those words to collect candidate Web snippets that may contain synonymous transliterations. After our preliminary experiments, three feasible strategies are proposed as follows.

Strategy 1 (Direct query strategy): A transliteration *TL* or its original foreign noun *ORI* is able to collect the target snippets. The strategies are denoted as Q_{TL} and Q_{ORI}. Given a Chinese transliteration, its foreign origin can be automatically determined (Hsu et al. 2007; Lee et al. 2006; Lin & Chen 2000; Lin & Chen 2002; Sakoe & Chiba 1978; Somers 1998; Stalls & Knight 1998; Tsuji 2002). Note when the Q_{ORI} is used, the search domain is set to be the Chinese Web pages only in order to retrieve Chinese transliterations.

Strategy 2 (Indirect query strategy): Indirect query strategy utilizes association words of the input *TL* to collect the target snippets. Denote $T = \{t_1, \ldots, t_k, \ldots, t_K\}$ be a set of top *K* association words obtained from the core snippets by the giving *TL*. A query set $Q_m\text{-}A_s = \{q_1, q_2, \ldots, q_n, \ldots, q_N\}$ is formed where *m* indicates the number of association words taken from T and *N* is the number of queries constructed where $N = C(K, m)$. For instance, assume the top 3 association words of 戈巴契夫 are $T = \{$冷戰 (cold war), 共產黨 (Chinese Communist Party), 民主 (democracy)$\}$, the query set $Q_{2\text{-}As}$ is then consisted of three elements, namely, $Q_{2\text{-}As} = \{q_1 =$ (冷戰, 共產黨); $q_2 =$ (共產黨,民主); $q_3 =$ (冷戰,民主)$\}$.

Strategy 3 (Integrated query strategy). An integrated query strategy $Q_{m\text{-}AsOri}$ which includes association words and the *ORI* is also considered, which helps improve the quality of the target snippets. For the example of 戈巴契夫, the set of $Q_{2\text{-}AsOri}$ includes elements $\{q_1 =$ (冷戰, 共產黨, Gorbachev); $q_2 =$ (共產黨, 民主, Gorbachev); $q_3 =$ (冷戰, 民主, Gorbachev)$\}$.

3.3 Candidate Synonymous Transliteration Extraction

After collecting the set of target snippets which may contain synonymous transliterations, we propose a procedure to detect them. The detection task has two major processes. The first is to acquire a set of *n*-gram terms, referred to as candidate synonymous transliterations or *CST*, with concerning the appropriate length of the transliteration (*TL*). The second is to discard, by a dynamic alignment technique, the *n*-gram terms which are not possible to be true synonymous transliterations.

3.1.1 *n-gram term segmentation*

We remove HTML tags and symbols in the target snippets and discard known words with the help of a Chinese dictionary since transliterations are unknown words to a regular dictionary. *N*-gram terms are then segmented from the remaining text. The length of *TL* and its *ST* may be different due to different translators. According to our observation, most of synonymous transliterations have the same length but some have a discrepancy of 1. Consequently, we segment the text to *n*-gram terms where $n = |TL|\text{-}1$ to $|TL|+1$ and $n \geq 2$.

3.2.2 *Dynamic Alignment between TL and n-gram terms*

Segmented *n*-gram terms contain a lot of false positives, i.e., not true synonymous transliterations. To determine whether an *n*-gram term is a *ST* or not, we use the dynamic warping function (Hsu et al. 2007; Kuo et al. 2007) and justify the boundary to aid in aligning between a *TL* and an *n*-gram term. The idea is that the corresponding first and last characters of the *TL* and an *ST* are usually the same or highly similar in pronunciation, for instance, 戈巴契夫 (ge ba qi fu), 哥巴卓夫 (ge ba zhuo fu), 戈爾巴喬夫 (ge er ba qiao fu). An *n*-gram term which does not match well at the first and the last characters with the *TL* is probably not a true *ST*, for instance, 理哥巴卓 (li ge ba zhuo) and 戈巴契夫 (ge ba qi fu).

Two cases of equal length and unequal length have to be considered. Moreover, an exception is also identified and handled to improve the quality of determining *CSTs*.

Equal length

The result of dynamic alignment is as shown in Fig. 2 by the example of *TL* 戈巴契夫 (ge ba qi fu) and its three equal-length 4-gram terms, 哥巴卓夫 (ge ba zhuo fu), 理哥巴卓 (li ge ba zhuo) and 巴卓夫寫 (ba zhuo fu xie). The bevel arrow anchor represents the alignment of two Chinese characters and the sequence of the bevel arrows represents the best path for matching the two terms. We can observe that only the sequence 哥巴卓夫 (ge ba zhuo fu) satisfies the constraint of matching at the first and the last character with *TL*.

	哥(ge)	巴(ba)	卓(zhuo)	夫(fu)
戈(ge)	↖			
巴(ba)		↖		
契(qi)			↖	
夫(fu)				↖

(a)

	理(li)	哥(ge)	巴(ba)	卓(zhuo)
戈(ge)		↖		
巴(ba)			↖	
契(qi)				↖
夫(fu)				↑

(b)

	巴(ba)	卓(zhuo)	夫(fu)	寫(xie)
戈(ge)	↖			
巴(ba)		↖		
契(qi)		↑		
夫(fu)			↖	←

(c)

Figure 2: Dynamic alignment between a *TL* with its three *n*-gram terms.

Unequal length

The length of the *n*-gram term and the *TL* is not equal due to a syllable of the original term is ignored in transliterating process. The place of the ignored character for a transliteration usually happens in the middle or the suffix but seldom happens in the prefix. In other words, comparing a *TL* to its *ST*, the surplus character is often in the middle or in the suffix of the longer length of the *TL* (or *ST*).

If the surplus character happens in the middle, the above matching constraint still applies. For instance, among the three 5-gram terms 戈爾巴喬夫 (ge er ba qiao fu), 在戈爾巴喬 (zai ge er ba qiao), and 爾巴喬夫和 (er ba qiao fu he), only the first one is a true synonymous transliteration. As shown in Fig. 3(a), 戈爾巴喬夫 matches with 戈巴契夫 at the first and the

last character.

	戈 (ge)	爾 (er)	巴 (ba)	喬 (qiao)	夫 (fu)
戈(ge)	↖	←			
巴(ba)			↖		
契(qi)				↖	
夫(fu)					↖

(a)

	在 (zai)	戈 (ge)	爾 (er)	巴 (ba)	喬 (qiao)
戈(ge)		↖	←		
巴(ba)				↖	
契(qi)					↖
夫(fu)					↑

(b)

	爾 (er)	巴 (ba)	喬 (qiao)	夫 (fu)	和 (he)
戈(ge)	↖				
巴(ba)		↖			
契(qi)			↖		
夫(fu)				↖	←

(c)

Figure 3: Dynamic alignment between a *TL* and its three *n*-gram terms with a different length from the *TL*.

Extra-character exception

The other case is that the surplus character of a true synonymous transliteration happens in the suffix of the longer one, such as the examples of 姆(mu), 德(de) and 兒(er) in Fig. 4. This indicates that the ignored character with corresponding to its place appearing in the suffix have to be tackled with care.

	貝(bei)	克(ke)	漢(han)	姆(mu)
貝(bei)	↖			
克(ke)		↖		
漢(han)			↖	←

(a)

	倫(lun)	斯(si)	斐(fei)	德(de)
倫(lun)	↖			
斯(si)		↖		
斐(fei)			↖	←

(b)

	費(fei)	雪(xue)	兒(er)
費(fei)	↖		
雪(xue)		↖	←

(c)

Figure 4: Some suffix phones usually are ignored, resulting in an extra-character in the alignment.

In order to solve such kind of problem, an *extra-character exception* is established as follows. If the last second character of the *n*-gram term is matched with the last character of the *TL* and the last character of the *n*-gram term is one of the syllables which might often be ignored, the *n*-gram term is then considered a true synonymous transliteration, like the three examples in Fig. 4.

Those syllables which might be ignored include “m”, “er”, “d” and “t” and they are ususally transliterated respectively as 姆 (mu), 兒/爾 (er), 德 (de), and 特 (te). Moreover, such English phonemes as “s” and “z” usually transliterated to various Chinese characters, such as 茲 (zhi), 池(chi), 仕 (shi), 日 (ri), 子 (zi), 慈 (ci), 斯(si) should also be considered. Each of which Mandarin phonetic symbol is coming from the consonant group either retroflex or dental.

4. RECOGNIZE SYNONYMOUS TRANSLITERATIONS

Dynamic alignment mentioned in the previous section help to eliminate some false positive *n*-gram terms. There are still a lot of remaining *n*-gram terms, referred to as *candidate synonymous transliterations or CST*. Further processes are needed to rank the remaining terms so as the true *ST* could rank higher than those noise terms.

We construct a decision model for the ranking task, which consists of two major components. First, we utilize a digitized sound comparison technique, namely, *Character Sound Comparison* (CSC) (Hsu et al. 2007) which will give a high score to a *CST* sound similar to the TL. Second, we take into account semantic similarity between the *CST* and the TL, referred to as *Context Matching* (CM) which will give a high score to a *CST* semantically similar to the *TL*. The final ranking takes into consideration both similarities.

The decision model is illustrated in Fig. 5 where a *CST* 戈爾巴喬夫 is being handled with respect to the *TL* 戈巴契夫. In the first phase, we use the CSC approach to measure the sound similarity score and then check whether it is larger than threshold θ. In this case, the *CST* (戈爾巴喬夫) is retained because its similarity score with *TL* 戈巴契夫 exceeds the threshold. The second phrase is to download several Web snippets for deriving semantics of the *CST*. At the end, the ranking of the *CST* 戈爾巴喬夫 to the *TL* 戈巴契夫 against other *CSTs* is determined by aggregating the CSC and the CM score. The following sections detail how the CSC and the CM score are obtained.

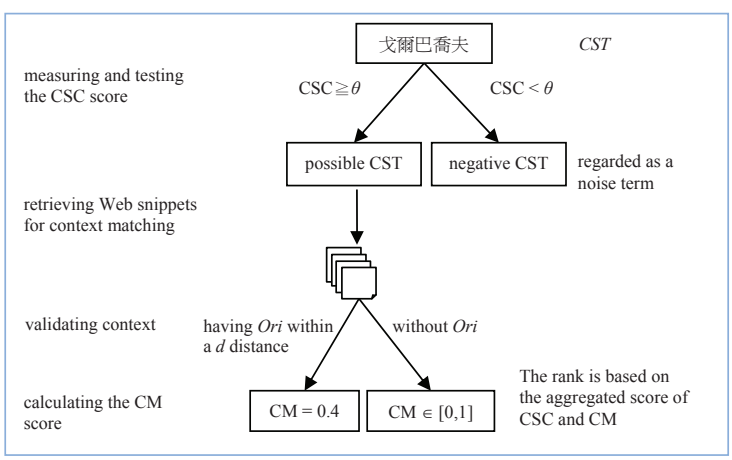

Figure 5: Decision model for ranking a candidate synonymous transliteration.

4.1 Chinese Sound Confirmation

A transliteration usually has pronunciation close to their original foreign noun. Therefore, synonymous transliterations usually sound similarly to each other, such as the transliterations

of Gorbachev “戈巴契夫” and “戈爾巴喬夫”. Based on this observation, we measure the pronunciation similarity between a *CST* and the *TL*.

Specifically, we use the Chinese Sound Comparison method (CSC) (Hsu et al. 2007) which compares two words by their digitalized physical sounds. The CSC has advantages over grapheme-based and phoneme-based approaches due to embedding more discriminative information in the digitalized sound signals. Grapheme-based approaches are mainly based on the number of identical alphabets in the two comparing words. Phoneme-based approaches (Chen et al. 2006; Kondrak 2003) are mainly based on the pronunciation similarity between phones. The similarity scores between phones are assigned by some predefined rules which take articulatory features of phones into consideration.

CSC is based on dynamic programming to compare digitalized Chinese character (or Hanzi) sounds. In particular, given two Chinese terms $A = \{a_1 a_2 \ldots a_N\}$ and $B = \{b_1 b_2 \ldots b_M\}$ where a_n is the n^{th} character in Chinese term A and b_m is the m^{th} character in Chinese term B. N may not be equal to M. A dynamic programming based approach to compare the similarity of smallest distortion for A and B via aiming at adjusting the warp on the time axis will be employed in this paper. The recurrence formula is defined as follows,

$$T(n,m) = \max \begin{cases} T(n-1,m-1) + sim(a_n, b_m) \\ T(n-1,m) \\ T(n,m-1) \end{cases} \quad (2)$$

where $T(N, M)$ is the similarity score between $\{a_1 a_2 \ldots a_N\}$ and $\{b_1 b_2 \ldots b_M\}$, and $sim(a_n, b_m)$ is the similarity score between the two Chinese characters.

We constructed two similarity matrices for comparing the similarity between two Chinese words, one of 37 phonetic symbols and the other of 412 basic characters. The construction procedure includes two main steps. First, features of each sound are extracted via a sequence of speech processing techniques, including frame segmentation, endpoint detection, and frequency transformation. The result is a feature vector of dimension 26, consisting of 12 cepstral, 12 delta-cepstral coefficients, energy, and delta-energy in the *MFCC* domain. Second, after transforming the speech sound to sound vectors, similarity between a pair of sounds was measured so as to acquire the two similarity matrices, which includes the 37 phonetic notation matrix and the 412 Chinese character sound matrix. The 412 basic character sound matrix includes all Chinese character pronunciations without considering the four tones.

According to our experience, final sound heavily influences speech sound comparison. Faced with this problem, we adopted an initial-weighted comparison approach, which involved a balancing adjustment: weighting the initial consonants of the characters to balance the bias caused by the final sounds. The 37 phonetic symbol similarity matrix is used to provide the similarity data between the initials of the characters.

$$sim(a_n, b_m) = w \times sim_{s37}(a_n.IC, b_m.IC) + (1-w) \times sim_{s412}(a_n, b_m) \quad (3)$$

$a_n.IC$ and $b_m.IC$ represent their initial consonant *(IC)*. $sim(a_n, b_m)$ is the weighted similarity between the character an and bm obtained from the similarity matrices of the 37 phonetic symbols and 412 character pronunciations. For example, both sim_{s37}(ㄙ, ㄕ) and sim_{s403}(ㄙㄣ, ㄕㄥ) have high scores due to high similarity in their speech sounds. The parameter w represents a trade-off measure between the initial consonant and the whole character, which is set to 0.4 empirically (Hsu et al. 2007). The CSC score between *A* and *B* is then defined by Eq. (4),

$$S_{CSC}(A,B) = \frac{T(N,M)}{(N+M)/2} \quad (4)$$

where *N* and *M* are the lengths of two Chinese terms *A* and *B*. The choice of normalization operation significantly influences the similarity comparison. We set to it to the average length of *N* and *M* (Hsu et al. 2007).

4.2 Context Matching

An unrelated *n*-gram term may just happen to sound similarly to the *TL*. Therefore, context matching is employed to further improve the ranking of a true *CST*. The idea behind context matching is that the meaning of a *TL* shall be similar to that of its *ST*; Thus if the context of a *CST* is similar to that of a *TL*, the *CST* shall be given a higher rank. In other words, if the Web snippets retrieved by a *CST* are similar to the snippets retrieved by the *TL*, the *CST* and the *TL* are considered highly matched in terms of context.

Denote $AW_{TL} = \{t_k\}$ for k = 1, 2, ..., |AW| be the feature set extracted from the set D_{TL} of snippets retrieved by the *TL*. It is worth noting that we can use the association words of the *TL* obtained in the early stage from the core snippets as the feature set so that no additional computation is required. Denote $D_{CST} = \{d_1, ..., d_j, ...d_J\}$ be the set of Web snippets retrieved by the *CST* and AW_{CST} be the feature set extracted from D_{CST}. The more common association words in the two sets AW_{TL} and AW_{CST}, the more semantically similar the *CST* and the *TL* are. The semantic similarity between *TL* and *CST* is calculated by the cosine measure like Eq. (5). We include a weight $0 \le \alpha \le 1$ for experimentally exploring its impact.

$$Sem_{AW}(TL, CST) = \alpha \times \frac{AW_{TL} \cdot AW_{CST}}{\| AW_{TL} \| \| AW_{CST} \|} \quad (5)$$

From our experience, whether the original foreign word of a *TL* appears in the set of retrieved snippets play a key role in the decision. We therefore design an formula, namely $S_{sem}(TL, CST)$, to take the presence of the original word into consideration. It is to assign a

pre-determined similarity score if D_{CST} contains the original foreign word or the weighted cosine score otherwise. The pre-determined similarity score is set to 0.4 in our experiment. It is determined by the following statistic analysis. Among the set of *CSTs* with their *CSC* score no less than 0.6 and the original word occurring in the set of returned snippets. 140 of the 368 are true synonymous transliterations, which represents a probability close to 0.4. Moreover, we observe the distance between the *CST* and the foreign word is also an important factor. The closer between the two, the more likely the *CST* is an ST. The formula is then defined by Eq. (6).

$$S_{sem}(TL, CST) = \begin{cases} 0.4, & \text{if } \min_{d_j \in D_{CST}} \{| dist_{d_j}(CST, Ori) |\} \le d \\ Sem_{AW}(TL, CST), & \text{otherwise} \end{cases} \quad (6)$$

$| dist_{d_j}(CST, Ori) |$ returns a distance value between the *CST* and the *Ori* if the original foreign noun *Ori* is included in $dj \in D_{CST}$ and otherwise the cosine score. The distance is measured by counting the number of Chinese characters. English words and delimiters such as punctuations, bracket and spaces are ignored in the distance calculation. Note that *Ori* could occur before or after the *CST*.

Finally, similarity between a *TL* and a *CST* which takes into consideration pronunciation similarity and Web-page context is a weighted linear combination of the CSC score S_{CSC} and the semantics score S_{Sem}. The equation is defines as follows.

5. EXPERIMENTS

5.1 Experimental Data

Two datasets are used for the experiments. The first dataset, referred to as D50 (Hsu et al. 2007), includes a total of 50 Chinese transliterations (TLs) collected from the Web as shown in Table 1. Their length is 2, 3 or 4, which are most commonly seen in Chinese transliterations. The number of transliterations in each group is 10, 30 and 10, respectively.

Table 1：Fifty Chinese transliterations used for the experiments

TL	Ori	TL	Ori	TL	Ori
布希 (bu xi))	Bush	柯林頓 (ke lin dun)	Clinton	羅伯茲 (luo bo zi)	Roberts
費雪 (fei xue)	Fisher	迪士尼 (di shi ni)	Disney	所羅門 (suo luo men)	Solomon
蓋亞 (gai ya)	Gaea	加奈特 (jia nai te)	Garnett	柴契爾 (chai qi er)	Thatcher
蓋茲 (gai zi)	Gates	赫爾利 (he er li)	Hurley	托拉斯 (tuo la si)	Trust
胡笙 (hu sheng)	Hussein	傑克遜 (jie ke xun)	Jackson	華勒沙 (hua le sha)	Walesa
詹森 (zhan sen)	Jansen	哈米尼 (ha mi ni)	Khamenei	溫絲蕾 (wen si lei)	Winslet
喬登 (qiao deng)	Jordan	路希奧 (lu xi ao)	Lucchino	阿米塔吉 (a mi ta ji)	Armitage
奈米 (nai mi)	Nano	曼德拉 (man de la)	Mandela	賽普拉斯 (sai pu la si)	Cypress
鮑爾 (bao er)	Powell	馬怪爾 (ma guai er)	McGuire	戈巴契夫 (ge ba qi fu)	Gorbachev
雪梨 (xue li)	Sydney	納亞夫 (na ya fu)	Najaf	喀爾巴拉 (ke er ba la)	Karbala
亞馬遜 (ya ma xun)	Amazon	歐尼爾 (ou ni er)	O Neal	奈西利亞 (nai xi li ya)	Nasiriyah
雅典娜 (ya dian nuo)	Athena	皮爾遜 (pi er xun)	Pearson	倫斯斐德 (lun si fei de)	Rumsfeld
巴薩拉 (ba sa la)	Basra	裴洛西 (pei luo xi)	Pelosi	史瓦辛格 (shi wa xin ge)	Schwarzenegger
貝克漢 (bei ke han)	Beckham	佩雷斯 (pei lei si)	Peres	史柯西斯 (shi ke xi si)	Scorsese
布萊爾 (bu lai er)	Blair	比卡丘 (bi ka qiu)	Pikachu	魏克菲爾 (wei ke fei er)	Wakefield
布雷默 (bu lei mo)	Bremer	篷比杜 (peng bi du)	Pompidou	伍夫維茲 (wu fu wei zi)	Wolfowitz
巴非特 (ba fei te)	Buffett	歐萊禮 (ou lai li)	Reilly		

The second dataset, referred to as D97, is from the 2008 TIME 100 list of the world's most influential people (Time 2009). There are a total of 104 names in the list since there are four entries of which each includes two names. Ninety seven names are retained for experiment. Seven names are ignored which include Ying-Jeou Ma, Jintao Hu, Jeff Han, Jiwei Lou, Dalai Lama, Takahashi Murakami, and Radiohead. The first four have a Chinese last name which has a standard Chinese translation. “Dalai Lama”, the spiritual leader of Tibet, also has a standard Chinese translation. The sixth one is a Japanese name of which translation is usually not by the use of transliterating. The last one is the name of a music band of which translation to Chinese is not according to its pronunciation but its meaning.

Most of the names in D50 are popular. Compared to those in D50, quite a few of the names in D97 are less popular to Chinese community and hence they appear in an expectedly limited number of Chinese Web pages. Some of them appeared in Chinese Web pages only after they were selected in the TIME 100 list.

5.2 Experimental Design and Evaluation Metrics

There are four major testing tasks in the experiment. Since we expect to mine synonymous transliterations from the Web snippets by the use of a search engine, we first determine several possible query strategies to collect the relevant target snippets from Google search engine, and then we evaluate how effective each query strategy is able to collect a better set of target snippets, which shall contain as many synonymous transliterations as possible. Second, we explore how many target Web snippets are enough to obtain synonymous transliterations via a search engine. Third, we test the effectiveness of the extracted candidate synonymous

transliterations (*CSTs*) by the help of CSC measure with dynamic alignment on several constraints. Finally, we investigate the quality of the decision model for the ranking task, which consists of the use of CSC and CM together. The results by the *baseline approach* which uses only the CSC score, does not consider context information, is also included for comparison.

The performance of the ranking task is evaluated via various measures including AR (average rank), ARR (average reciprocal rank) and inclusion rate, which are commonly used for performance evaluation in information retrieval, are calculated in this study according to the rank of similarity score of a true *ST* to the *TL*. ARR puts more weight on top-ranked terms. Assume *S* is the set of synonymous translations of the *TL* appear in the set of *n*-gram terms, R_{ST} is the score rank of an *ST* compared to the other *CSTs* of the *TL*, and TOP_n is the set of *CSTs* each of which has a CSC rank within top n. The measures are calculated as the following, $AR = 1/|S| \times \sum_{ST \in S} R_{ST}$, $ARR = \sum_{ST \in S} (1/RST)$ *and* $IR_n = 100\% \times 1/|S| \times \sum_{ST \in TOP_n}\{1\}$.

5.3 Performance of Query Strategies

Several feasible query strategies are used to collect target snippets from Google search engine. The performance of query strategies is evaluated with respect to how well they can retrieve the target set of candidate snippets which may contain synonymous transliterations. In practice, each of the *TLs* in both D50 and D97 is submitted to Google search engine and the first 20 snippets are collected as the *core snippets* of the *TL*, as discussed in Section 3. For each *TL*, the top five association words are used to collect various sets of the target snippets according to different strategies mentioned in Section 3.2. The numbers of the core snippets and top association words are determined empirically in the current study. Many users browse no more than the first two pages of snippets which are about 20 snippets. A larger size of association words would be used but demand extensive computation. To determine the optimal parameter setting requires further study in the future.

The retrieval methods used for comparison are listed below.

Q_{TL}: collecting snippets by using the *TL*;

Q_{Ori}: collecting snippets by using the original foreign word;

$Q_{m\text{-}As}$: collecting snippets by *m* association words for each query;

$Q_{m\text{-}AsOri}$: *collecting snippets by m association words plus the Ori for each query*;

Q_{GR}: Google's recommendation.

The first retrieval methods are described in detail in Section 3.2. Google occasionally suggests with respect to user queries synonymous transliterations, placed in the first or the last line of the first returned page. We therefore consider their recommendation as well in the experiment, as indicated in the list by Q_{GR}. The following presents the experimental results.

The total number of synonymous transliterations for D50 in the total returned snippets by all the strategies is 342, which is 6.8 on average per input *TL*. In D97, the total number of

synonymous transliterations in the total returned snippets by all the strategies is 401, which is 4.1 on average per input *TL*. The statistics is shown in Table 2.

Table 2：Performance of various retrieval methods

D50											
Retrieval method	**3-AsOri**	**2-AsOri**	**1-AsOri**	**3-As**	**Ori**	**4-As**	**2-As**	**4-AsOri**	**TL**	**1-As**	**GR**
Occurrence Prob.	0.86	0.90	0.86	0.56	0.52	0.30	0.38	0.18	0.34	0.06	0.04
No. of retrieved ST	259	242	154	65	37	36	35	34	21	3	1
Avg of retrieved ST	5.18	4.84	3.08	1.30	0.74	0.72	0.70	0.68	0.42	0.06	0.10
Uniqueness	50	30	11	8	4	1	1	2	3	1	0
Recall (out of 342)	0.76	0.71	0.45	0.19	0.11	0.11	0.10	0.10	0.06	0.01	0.003
D97											
	3-AsOri	**2-AsOri**	**1-AsOri**	**2-As**	**3-As**	**Ori**	**4-As**	**TL**	**4-AsOri**	**1-As**	**GR**
Occurrence Prob.	0.84	0.74	0.73	0.32	0.29	0.32	0.28	0.20	0.07	0.08	0.07
No. of retrieved ST	292	227	217	62	61	41	41	28	12	10	7
Avg of retrieved ST	3.01	2.34	2.24	0.64	0.63	0.42	0.42	0.29	0.12	0.10	0.07
Uniqueness	47	34	34	2	5	1	8	3	0	2	0
Recall (out of 401)	0.73	0.57	0.54	0.15	0.15	0.10	0.10	0.07	0.03	0.02	0.02

The *occurrence probability* means that at least one synonymous transliteration occurred in the set of returned snippets under the employed strategy. For example, forty-three (86%) out of fifty transliterations can retrieve at least one synonymous transliteration by using Q3-AsOri method in D50. The $Q_{3\text{-}AsOri}$ retrieves a total of 259 and 292 *STs (No. of retrieved ST)* for D50 and D97, respectively. On average, the strategy can retrieve 5.18 and 3.01 *(Avg of retrieved ST)* for a given transliteration for D50 and D97, respectively. For *uniqueness* which indicates how many *STs* which are retrieved uniquely by the method but not by the other methods, $Q_{3\text{-}AsOri}$ also achieve the best compared to the others. Based on the total number of synonymous transliterations retrieved together by all the methods, the results indicate that the $Q_{3\text{-}AsOri}$ method has the best *recall* rate over the other retrieval methods in both D50 and D97 datasets. The total recall rate of the top three methods together (i.e., $Q_{3\text{-}AsOri}$, $Q_{2\text{-}AsOri}$ and $Q_{1\text{-}AsOri}$) for D50 is 0.932, while that for D97 is 0.925.

The method of using the original word alone (which is the Q_{Ori} strategy) does not yield good recall. For instance, the given *TLs*, such as 費雪 (Fisher), 蓋亞 (Gaea), 鮑爾 (Powell), 巴薩拉 (Basra), and 賽普拉斯 (Cypress) of D50, have no STs in the retrieved snippets by Q_{Ori} but they do have by $Q_{3\text{-}AsOri}$. There are two main reasons. First, each of the returned snippets has only 3 lines of text (one line of anchor text and two lines of excerpt). Many of them do not contain any transliteration of the original word in that limited size of snippet text. Even though the search domain is set to only Chinese Web pages, many snippets contain only excerpted English text and no Chinese. Second, for those which do contain transliterations, lots of the transliterations in the set of returned snippets are identical. Moreover, some individual transliterations are very popular such that the returned snippets by the Q_{Ori} method contain only

that transliteration and no others. *A stricter* query strategy which additionally include association words along with the original foreign word help to bring the Web snippets containing various synonymous transliterations to the set of the returned first 1,000 pages.

In summary, experimental results indicate that including the foreign word along with their association words in the query outperforms those which do not include the original foreign word. Furthermore, the parameter m (the number of association words in a query) is better not to be greater than 4. Requesting too many association words in a snippet will limit the number of snippets that we can retrieve.

5.4 How many Web snippets are enough

This section explores how many target Web snippets are enough to obtain synonymous transliterations via a search engine. Although search engines, such as Google, Yahoo etc. give the count of returning pages much larger than 1,000, users can directly access the first 1,000 pages (Oyama et al. 2004). How do the synonymous transliterations distribute among the returned snippets? We analyze the collected Web snippets via the $Q_{3\text{-}AsOri}$ approach because $Q_{3\text{-}AsOri}$ can retrieve the most amounts of *STs*. Table 3 shows the inclusion rates of obtaining *STs* under top N returned snippets for D50 and D97. The result indicates 82.63% and 83.90% of *STs* are included in top 10 returned snippets for D50 and D97; 97.68% and 98.97% of STs, in top 100 returned snippets for D50 and D97, respectively. The information is valuable when computation complexity matters with respect to the recall rate.

Table 3：Performance of various retrieval methods

D50													
top *N*	1	3	5	10	20	30	40	50	100	200	300	500	1000
accumulated amounts	125	168	186	214	229	239	248	251	253	255	257	258	259
accumulated percentage (%)	48.26	64.86	71.81	82.63	88.42	92.28	95.75	96.91	97.68	98.46	99.23	99.61	100
D97													
top *N*	1	3	5	10	20	30	40	50	100	200	300	500	1000
accumulated amounts	155	192	211	245	259	267	276	285	289	290	291	291	292
accumulated percentage (%)	53.08	65.75	72.26	83.90	88.70	91.44	94.52	97.60	98.97	99.32	99.66	99.66	100

5.5 Performance of Synonymous Transliterations Extraction

This section presents how well the CSC measure with dynamic alignment on several constraints can help to extract candidate synonymous transliterations. Again we use the set of target snippets retrieved by the $Q_{3\text{-}AsOri}$ method as the experimental dataset.

The result in Table IV reveals several valuable points. First, setting a CSC score to 0.6 is a good choice which could significantly reduce the number of n-gram terms from 385,146 to 32,292 in D50 and from 114,857 to 12,831 in D97. The size of false positives is reduced to

8.3% and 10.9%, while most of true positives (99.2% and 100%) are retained in D50 and D97, respectively. Second, dynamic alignment is effective to further eliminate false positives to 4.5% and 5.8%; nevertheless some true positives are removed at the same time (95.4% and 87.7% remain) for D50 and D97. Finally, the extra-last-character exception help to bring the recall rate back to 98.5% and 100% with a small increase in false positive for both datasets.

Table 4：The effectiveness of various constraints which help to eliminate less promising n-gram terms

D50									
constraints	TRUE Positive	FALSE Positive	Total	TRUE Positive	FALSE Positive	Total	TRUE Positive	FALSE Positive	Total
CSC ≥ 0	259 (100%)	384887 (100%)	385146	259 (100%)	384887 (100%)	385146	259 (100%)	384887 (100%)	385146
CSC ≥ 0.5, 0.6 or 0.7	259 (100%)	163947 (42.6%)	164206	257 (99.2%)	32035 (8.3%)	32292	249 (96.1%)	3826 (1.0%)	4075
+ alignment	247 (95.4%)	57323 (14.9%)	57570	247 (95.4%)	17269 (4.5%)	17516	242 (93.4%)	2321 (0.6%)	2563
+ extra last character	255 (98.5%)	60359 (15.7%)	60614	255 (98.5%)	17702 (4.6%)	17957	249 (96.1%)	2411 (0.6%)	2660
D97									
constraints	TRUE Positive	FALSE Positive	Total	TRUE Positive	FALSE Positive	Total	TRUE Positive	FALSE Positive	Total
CSC ≥ 0	292 (100%)	114565 (100%)	114857	292 (100%)	114565 (100%)	114857	292 (100%)	114565 (100%)	114857
CSC ≥ 0.5, 0.6 or 0.7	292 (100%)	51591 (45.0%)	51883	292 (100%)	12539 (10.9%)	12831	277 (94.9%)	2012 (1.8%)	2289
+ alignment	256 (87.7%)	19279 (16.8%)	19535	256 (87.7%)	6683 (5.8%)	6939	249 (85.3%)	1196 (1.0%)	1445
+ extra last character	292 (100%)	21498 (18.8%)	21790	292 (100%)	7015 (6.1%)	7307	277 (94.9%)	1270 (1.1%)	1547

5.6 Improving ranking by context

We evaluate the performance of the decision model for improving ranking of a *CST* by taking context information into account, which consists of the use of CSC and CM. The evaluation measures include average rank, average reciprocal rank and inclusion rate.

Extensive experiments have been conducted, in which the CSC score parameter w steps from 0.4 to 0.9 by 0.1, the semantic score parameter α is set to 0, 0.3, 0.5, 0.7 or 1. The approach includes the information of CSC and CM, as shown in Eq. (6). The results by the baseline approach which uses only the CSC score, does not consider context information is also included for comparison. In fact, it is equivalent to set w to 1. For the sake of clarity, we do not show the results for each setting. However, the illustrative results are enough to demonstrate the general tendency.

Fig. 6 shows that AR and ARR with respect to various weights w and α in both datasets D50 and D97. Experimental results indicate that the combination of larger weight w (i.e. $w = 0.8$, 0.9) on the CSC score and a smaller weight α (i.e. $\alpha = 0.3$, 0.5) on the semantic score yields

better performance.

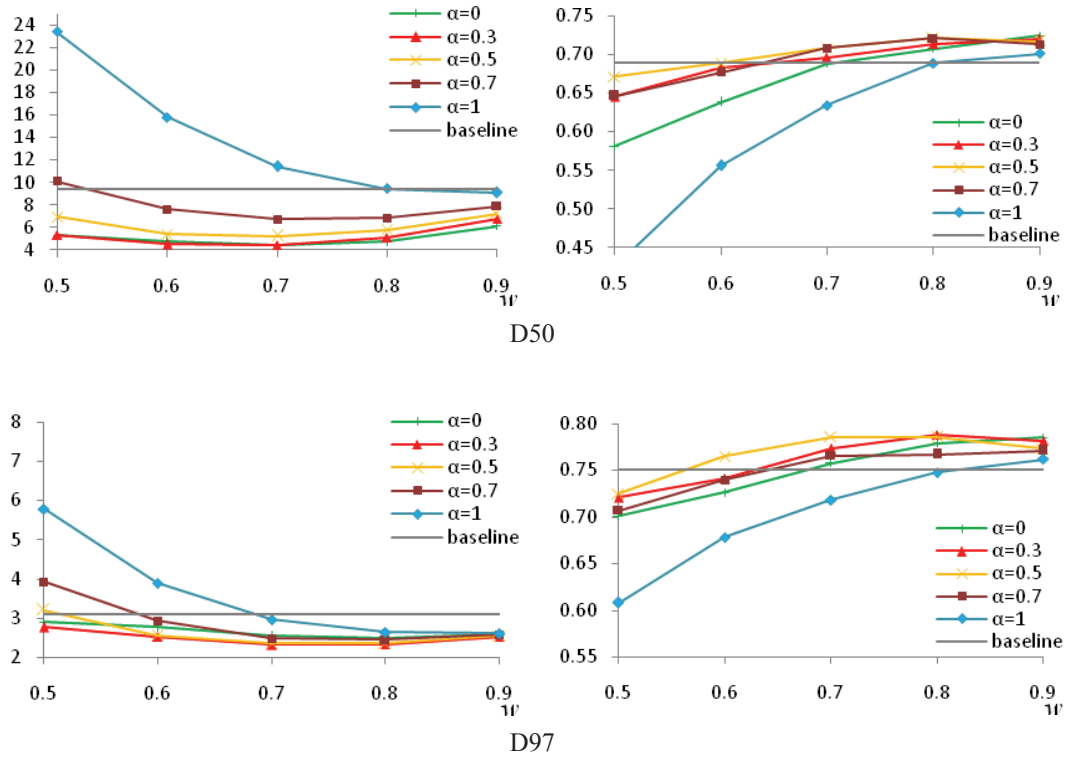

Figure 6: AR (left) and ARR (right) with various w and α.

We investigate whether the *Ori* of a *TL* appears in D_{CST} being useful to recognize the actual *ST* of the *TL*. This is analogy in that we compare the performance between Eq. (5) and Eq. (6). In general, the CM approach with considering the *Ori* information is better than that without considering the *Ori* information, as indicated in Fig. 7. The baseline approach which counts on only pronunciation similarity can be outperformed by the other two approaches, indicating the inclusion of context information help improving the performance. Moreover, the outcome implies that the foreign word, which is weighted by (1- α), might play an important role. This implication is conformed in the next section by another set of experiments in which snippets for context information are retrieved by using the *CST* plus the foreign word.

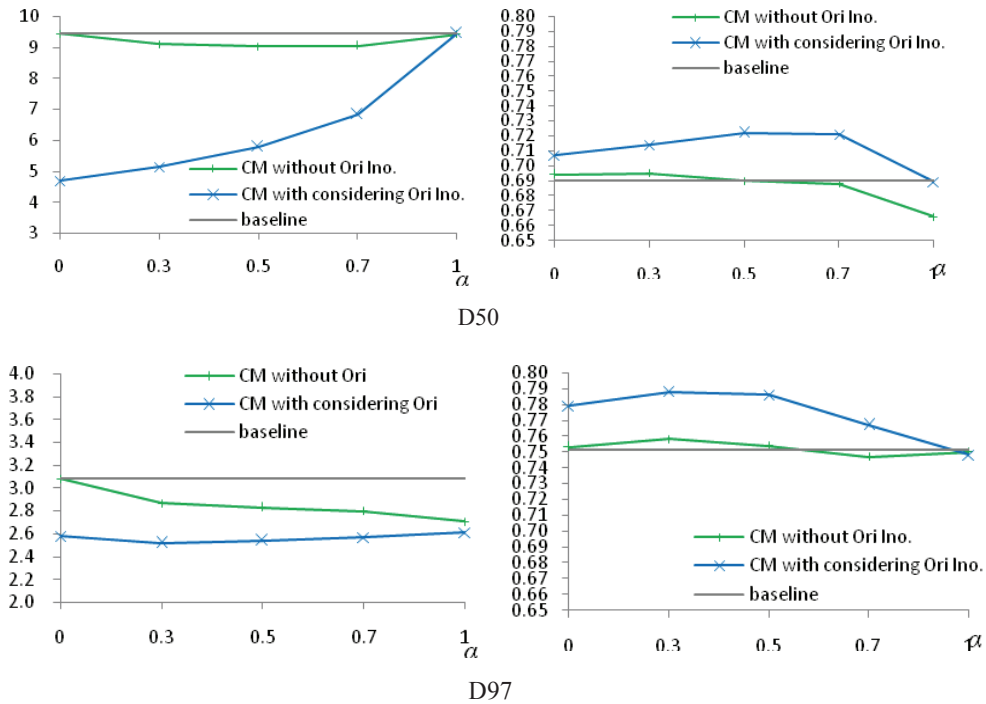

Figure 7: AR and ARR by CM approach with (without) considering *Ori* information under various α at w = 0.8.

Due to the important role that the foreign word plays in verifying whether a *CST* is indeed an *ST*, in the following experiments the snippets from which we derive context information are retrieved by using the *CST* plus the foreign word, contrast to the previous experiments in which the snippets are retrieved by using only the *CST*. This experiment is conducted for studying whether the distance between the *CST* and the *Ori* is helpful to recognize the actual *ST*.

We set α to 0.3 and 0.5, and we also set *w* to 0.8 and 0.9, according to the results in Fig.6. Moreover, we observe the distance between the *CST* and the *Ori* might also be a key factor. The distance parameter *d* is set to 0, 1, 2, 3, 4, 5 and no limitation (indicated by ∞). Recall that d is the number of Chinese characters in between the foreign word and the *CST*.

Experimental results indicate that distance parameter *d* shall be set to a small value between 0 and 2. As shown in Fig. 8, AR is the lowest and ARR is the highest when *d* is set to 0. The reason behind this is when a foreign proper noun appears with its transliteration in a document it is often nearby the transliteration.

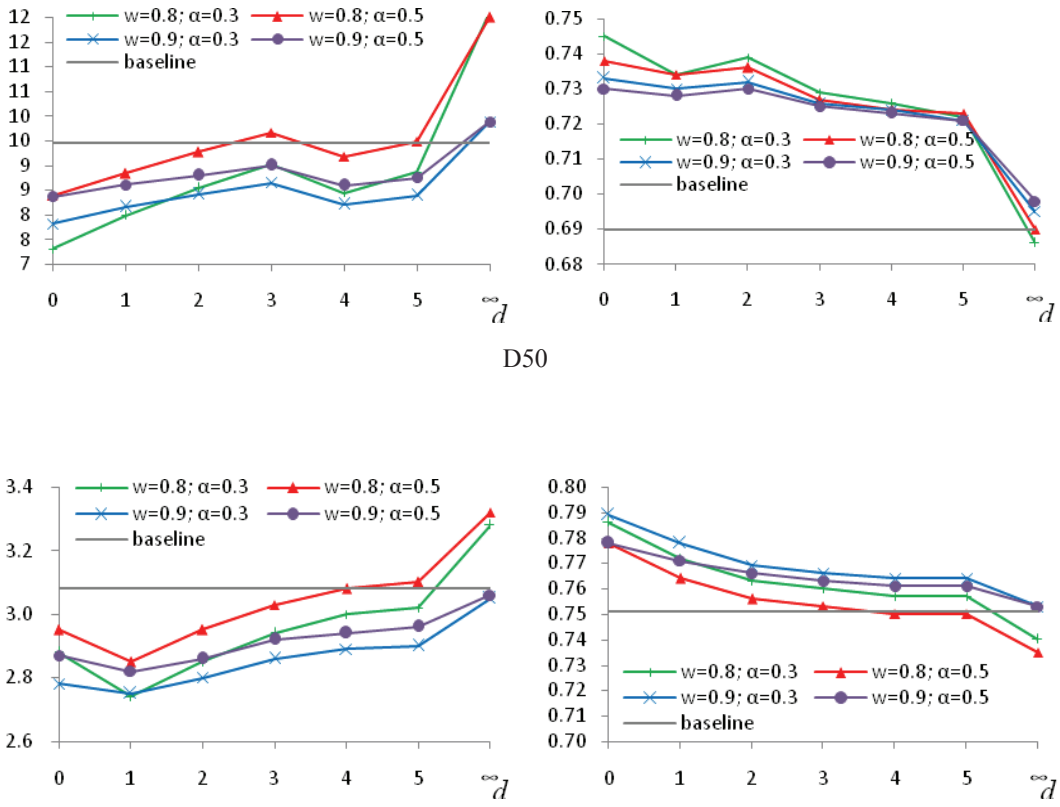

Figure 8: AR and ARR with various d.

Table V summarizes experimental results by showing the inclusion rates of two major configurations, the proposed method and the baseline. The baseline ranks a *CST* term based on only the phonetic CSC score while the proposed method takes context information into account. The presented results by the proposed method are under their best parameter setting ($w = 0.8$, $\alpha = 0.5$, and $d = 0$).

In both datasets D50 and D97, the baseline approach achieves modest performance which is able to include about 61.33% and 65.98% of synonymous transliterations in top 1 while the proposed method reaches 65.23% and 71.13%. There are several STs ranked outside the top 200 in D50, while the proposed method can include all the *STs* in top 40. That the results of the proposed method are better than those of the baseline indicates the inclusion of context information is able to improve performance.

Table 5：Inclusion rates of synonymous transliterations in top *n* candidate terms

dataset	method	Inclusion rates (%)									
		1	3	5	10	20	30	40	50	100	200
D50	the proposed	65.23	72.66	77.34	84.77	94.53	95.70	96.48	98.05	98.83	99.61
	the baseline	61.33	72.66	76.95	83.20	93.36	94.92	95.70	96.48	98.83	99.22
D97	the proposed	71.13	84.19	89.35	97.59	99.31	99.66	100	100	100	100
	baseline	65.98	80.76	87.29	94.85	97.59	98.63	99.66	99.66	100	100

6. CONCLUSIONS AND FUTURE WORK

In this paper, we first point out a critical issue in searching the Web involving transliterated foreign proper nouns, namely, the incomplete search-results problem resulting from the lack of translation standard on foreign proper nouns. To tackle the issue, we present a two-stage framework for mining as many synonymous transliterations as possible from Web snippets with respect to a given transliteration. The research result can be applied to construct a database of synonymous transliterations which can be used to expand an input query so as to alleviate the incomplete search problem resulting from the issue of different transliterations of a foreign word. Extensive experiments indicate that using association words plus the foreign word is preferred for collecting target snippets which may contain synonymous transliterations. In addition to the phonetic similarity score, the inclusion of context information helps to improve the ranking of synonymous transliterations against other noise terms. Regarding retrieving snippets for extracting context information of a candidate term, forming the query by using the candidate term plus the foreign word will yield better results in terms of determining whether the term is a synonymous transliteration.

Several aspects deserve future work. We have proved the feasibility of the proposed framework. Some of the adopted techniques for the components used in the framework are justified by referring to past work and some of them are justified by experiments in this study. It is possible to replace some components with other relevant techniques, such as using a training-based approach to generating possible transliteration candidates. Moreover, some of parameter settings are determined empirically in the present work, such as the numbers of core snippets and top association words. To determine the best techniques for individual components and the optimal parameter setting by quantitative measures require substantial efforts and extensive experiments. These issues can be further addressed in the future.

Acknowledgment

The work is supported by National Science Council, Taiwan under grant NSC 96-2410-H-224-004-MY2.

REFERENCES

1. AbdulJaleel, N., and Larkey, L.S. "Statistical transliteration for English-Arabic cross language Information retrieval," *Proceedings of the 12th international conference on information and knowledge management*, New Orleans, LA, USA, 2003, pp. 139-146.
2. Aggarwal, C.C., Al-Garawi, F., and Yu, P.S. "Intelligent crawling on the World Wide Web with arbitrary predicates," *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, 2001, pp. 96-105.
3. Al-Onaizan, Y., and Knight, K. "Machine Transliteration of Names in Arabic Text," *Proceedings of ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, Pennsylvania, 2002, pp.1-13.
4. Babaria, R., Nath, J.S., S, K., R, S.K., Bhattacharyya, C., and Murty, M.N. "Focused crawling with scalable ordinal regression solvers," *Proceedings of the 24th International Conference on Machine Learning*, Corvalis, Oregon, 2007, pp. 57-64.
5. Barbosa, L., and Freire, J. "An adaptive crawler for locating hidden Web entry points," *Proceedings of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007, pp. 441-450.
6. Brill, E., Kacmarcik, G., and Brockett, C. "Automatically harvesting Katakana-English term pairs from search engine query logs," *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, NLPRS, 2001, pp. 393-399.
7. Carpineto, C., Mori, R.d., ROMANO, G., and BIGI, B. "An information-theoretic approach to automatic query expansion," *ACM Transactions on Information Systems* (19:1), 2001, pp 1-27.
8. Chen, H.H., Huang, S.J., Ding, Y.W., and Tsai, S.C. "Proper name translation in cross-language information retrieval," *Proceedings of 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Quebec, Canada, 1998, pp. 232-236.
9. Chen, H.H., Lin, W., Yang, C.C., and Lin, W.H. "Translating/transliterating named entities for multilingual information access," *Journal of the American Society for Information Science and Technology* (57:5), March 2006, pp 645-659.
10. Cheng, P.-J., Teng, J.-W., Chen, R.-C., Wang, J.-H., Lu, W.-H., and Chien, L.-F. "Translating unknown queries with Web corpora for cross-language information retrieval," *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield, UK, 2004, pp.146-153.
11. Cilibrasi, R.L., and Vitanyi, P.M.B. "The Google similarity distance," *IEEE Transactions on Knowledge and Data Enginerring* (19:3), 2007, pp 370-383.
12. Collier, N., and Hirakawa, H. "Acquisition of English-Japanese proper nouns from noisy-

parallel newswire articles using Katakana matching," *Natural Language Pacific Rim Symposium* (NLPRS-97), Phuket, Thailand, 1997, pp. 309-314.

13. Connolly, J.H. "Quantifying target-realization differences," *Clinical Linguistics & Phonetics* (11:4), 1997, pp 267-298.
14. Fang, G., Yu, H., and Nishino, F. "Chinese-English term translation mining based on semantic prediction," *Proceedings of the COLING/ACL on main conference poster sessions*, Sydney, Australia, 2006, pp. 199-206.
15. Gao, W., Wong, K.-F., and Lam, W. "Phoneme-based transliteration of foreign names for OOV problem," *In First International Joint Conference on Natural Language Processing,* Sanya, Hainan, China, 2004, pp. 374-381.
16. Hsu, C.C., Chen, C.H., Shih, T.T., and Chen, C.K. "Measuring similarity between transliterations against noise data," *ACM Transactions on Asian Language Information Processing* (6:1), April 2007, pp 1-20.
17. Huang, S., Chen, Z., Yu, Y., and Ma, W.-Y. "Multitype features coselection for Web document clustering," *IEEE Transactions on Knowledge and Data Engineering* (18:4), 2006, pp 448-459.
18. Jeong, K.S., Myaeng, S.H., Lee, J.S., and Choi, K.S. "Automatic identification and back-transliteration of foreign words for information retrieval," *Information Processing and Management* (35:4), 1999, pp 523-540.
19. Knight, K., and Graehl, J. "Machine Transliteration," *Computational Linguistics* (24:4), 1998, pp 599-612.
20. Kondrak, G. "Phonetic alignment and similarity," *Computers and the Humanities* (37:3), 2003, pp 273-291.
21. Kuo, J.-S., Li, H., and Yang, Y.-K. "A phonetic similarity model for automatic extraction of transliteration pairs," *ACM Transactions on Asian Language Information Processing* (6:2), 2007.
22. Lee, C.J., Chang, J.S., and Jang, J.-S.R. "Alignment of bilingual named entities in parallel corpora using statistical models and multiple knowledge sources," *ACM Transactions on Asian Language Information Processing* (5:2), 2006, pp 121-145.
23. Li, H., Cao, Y., and Li, C. "Using bilingual Web data to mine and rank translations," *IEEE Intelligent Systems* (18:4), July/August 2003, pp 54-59.
24. Li, H., Zhang, M., and Su, J. "A joint source-channel model for machine transliteration," *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain, 2004, pp. 159-166.
25. Lin, W.H., and Chen, H.H. "Similarity measure in backward transliteration between different character sets and its applications to CLIR," *Proceedings of Research on Computational Linguistics Conference XIII*, Taipei, Taiwan, 2000, pp. 97-113.

26. Lin, W.H., and Chen, H.H. "Backward machine transliteration by learning phonetic similarity," *Proceedings of the Sixth Conference on Natural Language Learning*, Taipei, Taiwan, 2002, pp. 139-145.
27. Lu, W.H., Chien, L.F., and Lee, H.J. "Translation of Web queries using anchor text mining," *ACM Transactions on Asian Language Information Processing* (1:2), 2002, pp 159-172.
28. Lu, W.H., Chien, L.F., and Lee, H.J. "LiveTrans: Translation suggestion for cross-language Web search from Web anchor texts and search results," *Proceedings of Research on Computational Linguistics Conference XV* (ROCLING), 2003, pp.57-72.
29. Meng, H.M., Lo, W.K., Chen, B., and Tang, K. "Generating phonetic cognates to handle named entities in english-chinese cross-language spoken document retrieval," *Proceedings of the Automatic Speech Communication Recognition and Understanding Workshop*, 2001.
30. Och, F.J., and Ney, H. "A comparison of alignment models for statistical machine translation," *Proceedings of the 18th Conference on Computational Linguistics*, Saarbrücken, Germany, 2000, pp. 1086-1090.
31. Oh, J.-H., and Choi, K.-S. "An ensemble of transliteration models for information retrieval," *Information Processing and Management* (42:4), July 2006a, pp 980-1002.
32. Oh, J.-H., Choi, K.-S., and Isahara, H. "A machine transliteration model based on correspondence between graphemes and phonemes," *ACM Transactions on Asian Language Information Processing* (5:3), September 2006b, pp 185-208.
33. Oyama, S., Kokubo, T., and Ishida, T. "Domain-specific Web search with keyword spices," *IEEE Transactions on Knowledge and Data Engineering* (16:1), 2004, pp 17-27.
34. Oztekin, B.U., Karypis, G., and Kumar, V. "Expert agreement and content based reranking in a meta search environment using Mearf," *Proceedings of the 11th International Conference on World Wide Web*, Honolulu, Hawaii, USA, 2002, pp. 333-344.
35. Qin, J., Zhou, Y., and Chau, M. "Building domain-specific web collections for scientific digital libraries: a meta-search enhanced focused crawling method," *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, Tuscon, AZ, USA, 2004, pp. 135-141.
36. Sakoe, H., and Chiba, S. "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Signal Proc. ASSP* (26:1), 1978, pp 43-49.
37. Selberg, E., and Etzioni, O. "The metaCrawler architecture for resource aggregation on the Web," *IEEE Expert* (12:1), 1997, pp 11-14.
38. Somers, H.L. "Similarity metrics for aligning children's articulation data," *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1998, pp. 1227-1232.

39. Stalls, B.G., and Knight, K. "Translating names and technical terms in arabic text," *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*, Montreal, Quebec, Canada, 1998, pp. 34-41.
40. Tao, T., Yoon, S.-Y., Fister, A., Sproat, R., and Zhai, C. "Unsupervised named entity transliteration using temporal and phonetic correlation," *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney*, Australia, 2006, pp. 250-257.
41. "The 2008 Time 100: The World's most influential people," February 2009 (available online at http://www.time.com/time/specials/2007/0,28757,1733748,00.html?iid=redirect-time100).
42. Tsuji, K. "Automatic extraction of translational Japanese-Katakana and English word pairs from bilingual corpora," *International Journal of Computer Processing of Oriental Language* (15:3), 2002, pp 261-280.
43. Virga, P., and Khudanpur, S. "Transliteration of proper names in cross-lingual information retrieval," *Proceedings of ACL 2003 Workshop on Multilingual and Mixed-Language Named Entity Recognition*, Sapporo, Japan, 2003, pp. 57–64.
44. Wagner, R.A., and Fischer, M.J. "The string-to-string correction problem," *Journal of the Association for Computing Machinery* (21:1), 1974, pp 168-173.
45. Wu, J.C., and Chang, J.S. "Learning to find English to Chinese transliterations on the Web," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, 2007, pp. 996-1004.
46. Yoon, S.-Y., Kim, K.-y., and Sproat, R. "Multilingual transliteration using feature based phonetic method," *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, 2007.
47. Zhang, Y., and Vines, P. "Using the Web for automated translation extraction in cross-language information retrieval," *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, Sheffield, United Kingdom, 2004, pp. 162-169.