Computing and Informatics, Vol. 34, 2015, 795–822

A NEW CONFIRMATION FRAMEWORK FOR EXTRACTING SYNONYMOUS CHINESE TRANSLITERATION BASED ON PRONUNCIATION MODEL

Chien-Hsing Chen

Department of Information Management
Ling Tung University
No. 1, Ling tung Rd., Nantun, Taichung, Taiwan, R.O.C.
e-mail: ktfive@gmail.com

Chung-Chian Hsu

Department of Information Management
National Yunlin University of Science and Technology
No. 123, University Road, Section 3, Douliou, Yunlin, Taiwan, R.O.C.
e-mail: hsucc@yuntech.edu.tw

Abstract. There is no transliteration standard across all Chinese language regions, including China, Hong Kong, and Taiwan, and variations in Chinese transliteration have thus arisen in the process of transliterating foreign languages (English, for instance) into the Chinese language. In this paper, we propose an integrated confirmation framework to confirm a pair, that is, a transliteration and another term, whether it is synonymous. This framework includes two major steps. First, we study methods from several pronunciation-based approaches to measure similarity between Chinese characters; these approaches are specified for the comparison of Chinese synonymous transliterations. Second, we construct a new confirmation framework to confirm whether a pair of a Chinese transliteration and another Chinese term is synonymous. The presented framework is applied to extract synonymous transliteration pairs from a real-world Web corpus; this is valuable to build a new database of synonymous transliterations or support search engines so that they can return much more complete documents as Web search results to increase the usages in practice. Experiments show that our integrated confirma-

tion framework is effective and robust in confirming and extracting pairs of Chinese transliteration following the collection of synonymous transliterations from the Web corpus.

Keywords: Chinese transliteration variation, pronunciation-based approach, phonetic similarity, dynamic alignment, ensemble scheme

1 INTRODUCTION

There is no transliteration standard across all Chinese language regions; thus, many different Chinese transliterations can arise. For example, former Soviet President Gorbachev is translated into several different Chinese transliterations including (Gorbachev; ge ba qi fu), 哥巴卓夫 (ge ba zhuo fu) and 戈爾巴喬夫 (ge er ba qiao fu). The Australian city "Sydney" also produces different transliterations of 悉尼 (xi ni), 雪梨 (xue li) and 雪黎 (xue li). Someone who uses the Chinese language may never know all these different Chinese synonymous transliterations; hence, this level of Chinese transliteration variation leads readers to mistaken transliterated results, or to retrieve incomplete results when searching the Web for documents or pages if a trivial transliteration is submitted as the search keyword in a search engine such as Google or Yahoo. Moreover, while variations in Chinese transliteration have already emerged in all Chinese language regions, including China, Hong Kong and Taiwan, we still lack effective methods to address these variations. Most research focuses on machine transliteration across two different languages; in contrast, fewer efforts in the literature have focused on confirming a pair comprised of a Chinese transliteration and a Chinese term (i.e., proper name such as personal name and geographical name) as to whether it is synonymous.

In this paper, we propose an integrated confirmation framework for validating and extracting pairs from the Web, where each pair consists of a transliteration and another Chinese term. The framework considers a majority voting with multiple learning algorithms and a boosting scheme simultaneously. The returned result is either "synonymous" or "not synonymous" for the given pair. The contribution of this research is that the results of the confirmation framework can be applied to construct a new database of synonymous transliterations, which can then be used to increase the size of the transliterated vocabulary, making it useful to expand an input query in search engines such as Google and Yahoo. This can alleviate the problem of incomplete search results due to the existence of different transliterations of a single foreign word.

Two major steps are included in the framework for the sake of confirming whether a pair is synonymous. First, we study two Romanization transcription systems, including the National Phonetic System of Taiwan (BPMF system) and the Pinyin system, to transcribe Chinese characters into sound alphabets. Then, we adopt several pronunciation-based approaches to measure phonetic similarity be-

tween alphabets. The reasoning behind our approach is that the similarity resulted from evaluating the similarity based on pronunciation will be independent of a training corpus. In particular, we study a rule-based concept with different weights for the alphabets. In addition, we use a dynamic programming-based approach to obtain the similarity score for a given Chinese pair; that is, a Chinese transliteration versus another Chinese term.

Second, we cast the confirmation problem as an ensemble classification method specified on pronunciation-based models. The method is used to confirm whether a Chinese pair is synonymous or not, as it is a binary classification problem. We construct a classification learning algorithm with considering the use of a majority voting scheme and a boosting scheme [1] together to collect robustly pairs of Chinese synonymous transliteration from a real-world Web corpus. In addition, we also apply several weighted voting strategies [2] for classifiers. The presented method differs from a single classification-learning algorithm needed in boosting and bagging [3], or a majority voting scheme applied to combine multiple classification learning algorithms.

It is worth mentioning that, in practice, extracting synonymous transliterations from the Web corpus is a real-world problem and it is more valuable than doing the same from a prepared training corpus, because most transliterations are outside of usual Chinese vocabulary [4, 5]. We perform experiments on the Web search result snippets from the Google search engine by collecting documents from the Chinese-dominant Web and aiding in the extraction of transliteration synonyms. This paper is organized as follows. Following the introduction (Section 1), Section 2 illustrates related work. Section 3 shows an overall view of the integrated confirmation framework. Section 4 describes how to ascertain the phonetic similarity between a Chinese transliteration and another Chinese term using a pronunciation-based model. Section 5 defines a new ensemble framework for confirming a pair consisting of a Chinese transliteration and a segmented Chinese n-gram. Section 6 discusses experiments run on the real Web corpus. A conclusion and suggestions for possible future work are given in Section 7.

2 RELATED WORK

The framework for confirming whether a pair comprised of a Chinese transliteration and another Chinese term is synonymous includes two major steps. The first involves measuring the phonetic similarity for the pair. Second, we consider the confirmation problem as an ensemble classification task specified on pronunciation-based models.

2.1 Measuring Phonetic Similarity

Evaluating phonetic similarity between sound alphabets constitutes an open issue in the field of computational linguistics. A comparison of the sound alphabet in human languages in the process of machine transliteration is based on the phoneme and the

grapheme. A phoneme is the smallest pronunciation unit, whereas a grapheme is the written unit. Oh et al. identified three machine transliteration models, namely, the grapheme-based transliteration model, the phoneme-based transliteration model and the hybrid transliteration model [6]. Transliterating a name from one language into another language includes forward transliteration and backward transliteration [7]. The similarity evaluation between sound alphabets is critical. This can be evaluated using either a statistical-based model or a pronunciation-based model.

Statistical estimations for learning the similarities between sound alphabets have been applied to various applications in machine transliteration. The task in a statistical-based model requires an approximate probability distribution to be referred to a training corpus. Knight et al. proposed a backward phoneme-based transliteration system from Japanese to English consisting of five stages [7]. In their work, they proposed a tree-based structure consisting of weighted finite-state transducers (WFSTs) organized from a set of English-Japanese sound alphabet sequence based on probabilities and on Bayes' theorem for estimating similarity among phonemes. AbdulJaleel et al. proposed a statistical grapheme-based transliteration model for machine-learning of the Arabic alphabet and the English alphabet [8]. Virga et al. presented an application for a cross-lingual information retrieval between an English term and a Chinese transliteration based on statistical machine transliteration [9]. Li et al. studied a dynamic alignment process using a maximum likelihood estimation to handle the task of English-Chinese transliteration [10]. Wan and Verspoor [11] also introduced an algorithm for mapping English names to Chinese names. Oh et al. studied a machine transliteration model using three learning algorithms, including a maximum entropy model, a decision-tree model and a memory-based learning model [6]. Oh et al. conducted experiments on English-to-Korean transliteration and English-to-Japanese transliteration and described a ranking scheme for transliteration extraction from Web data [12]. Gao et al. employed a training process of dynamically-discovered alignments to handle English-to-Chinese machine transliteration [13]. Tao et al. built a cost matrix to learn the distance between sound alphabets via the dynamic alignment learning of substitution, deletion and insertion operations [14]. They conducted experiments on English-to-Arabic, English-to-Chinese and English-to-Hindi correspondences using the learned cost matrix. Simon et al. used the Levenshtein algorithm to measure the similarity between two Chinese terms so as to recognize the transliterated entity [15].

The pronunciation-based model takes advantage of phonetic features produced by tones and pronunciation locations, such as lip, palate, tongue or bilabial pronunciation. It learns human pronunciation sounds using phonetic features to calculate the similarity of phonetic segmental speech units. Connolly proposed a scheme for evaluating the similarity score between phonemes [16]. He identified two perceptual features in order to separate consonant phonemes into six groups. The similarity score of two consonants within the same group would be higher than that for those in different groups. Chen et al. argued that different sound alphabets may contain the same sounds and proposed a scheme to compare similarities between

Chinese and English personal names [17]. Lin et al. designed a scoring scheme for assigning similarity between phonemes [18]. They argued that the similarity score of a matched consonant sound alphabet is different from that of a matched vowel one. They also designed a set of corresponding pairs, such as the pairs (b, p) and (d, t) are designated as high similarity in comparison to phonemes containing them. Kondrak presented a scoring scheme for computing phonetic similarity between alphabets [19]. He argued that assigning equal weight to all features cannot address the problem of unequal relevance of features; thus, the scheme considered articulatory features and assigned different weights to different features. Hsu et al. proposed an approach to evaluate similarity between Chinese characters [20]. This approach was based on human pronunciation sounds corresponding to a Mandarin phonetic symbol tree; similarly, an alphabet that must be pronounced should be concerned with various pronunciation locations, such as the lip, palate, or tongue. The sound feature was thus extracted to form a vector with 26 dimensions via the process of extracting mel-frequency cepstrum coefficients (MFCC).

Accordingly, the pronunciation-based model focuses on learning phonetic features, whereas the statistical-based model focuses on estimating the probability distribution on a prepared training dataset. The fact that the pronunciation-based model is independent of the training corpus motivates us to learn several pronunciation-based approaches to aid in the comparison between Chinese transliterations.

2.2 Ensemble Classifier

Previous studies have shown that an aggregating ensemble scheme can improve performance in terms of classification accuracy, since there is no individual approach that will always yield the best result. The ensemble scheme has been studied in many research areas, including pattern recognition, information retrieval, data mining, and machine learning. In a classification task, the ensemble scheme determines a final class label from a set of individual results that are usually generated from a set of individual classifiers.

The majority-voting scheme works as an integrated task, allowing multiple classification learning algorithms to contribute corresponding votes. It then produces a winner vote from the results of the participating classifiers. The critical issue is how to determine which classification-learning algorithms are appropriate to use together to produce results. Moreover, two ensemble schemes, namely, bagging and boosting, have been used to numerous extensions and applications. The bagging scheme combines multiple independent classifiers to yield a prediction class label by integrating their corresponding votes [3]. It uses a bootstrapping strategy to generate subsets of training data. The generated subsets of training data are all learned according to a classification-learning algorithm so as to form a set of classifiers, each of which contributes a vote. Then, the votes generated from the set of classifiers are combined to produce the winning vote. The booting scheme is similar to bagging, except that their roles in generating subsets of training data differ. In boosting, each subset of training data is drawn with respect to the data distribution according to

training quality; thus, the data distribution of a new generation is different from the old one.

Accordingly, the advantage of the majority-voting scheme in multiple classification-learning algorithms is in classification stability [21, 22], whereas the advantage of the bagging and the boosting schemes lies in the improved performance in classification accuracy [23, 24]. In this paper, we propose a new ensemble framework to employ the majority-voting scheme and boosting together. To design such framework is because that the framework can collect pairs of Chinese synonymous transliteration from a real-world Web corpus with considering the performance in terms of stability and accuracy at the same time.

2.3 Contributions

The framework proposed in this paper differs from earlier works in several principal aspects. First, we study several pronunciation-based approaches to compare a pair consisting of a Chinese transliteration and another Chinese term. We do not need to prepare a predetermined dataset for similarity estimation between sound alphabets, as it is required in the work of the statistical-based approaches. Second, our framework is specified over an ensemble of several approaches to the confirmation of Chinese transliteration pairs. It integrates several pronunciation-based approaches for confirming pairs of synonymous transliteration. Third, the framework considers the use of the majority-voting scheme and the boosting scheme together to improve performance in terms of classification stability and accuracy. Finally, our framework confirms and extracts pairs of synonymous transliteration from a real-world Web corpus, which can then be used to retrieve more Web pages relevant to the query; thus, our research might bring great contributions in practice. For example, one may submit a transliteration to search engines such as Google and Yahoo, the results of the framework can be used to recommend its synonyms to the user for exploring other related Web snippets.

3 OVERALL FRAMEWORK

We propose an integrated confirmation framework to confirm and extract synonymous transliteration pairs comprised of a transliteration and its synonym. The framework is shown in Figure 1. Given a transliteration (TL), the study starts with a collection of Web snippets, since much research has begun to utilize abundant Web resources for various issues of cross-lingual information retrieval [25]. Usually, a transliteration co-occurs in proximity to its original English word within the same document [25]; hence, we collect Web snippets based on this assumption. Many studies are continuously used to extract information by using search keyword for a search engine instead of by visiting a sufficiently large Web page set from the Web. Of course, we can use a Web spider to crawl the set, this is, in general, done for many applications in the information retrieval field. In a contrast, to extract

information from the snippets responded from a search engine, crawling this set usually brings much time-consuming and computational load in handling the set. Therefore, we assign a given term which is submitted to a search engine so as to collect Web snippets.

Then, we produce a pair (a given term, a target term), where the given term is the *TL* we assign in advance and the target term means an n-gram which is segmented from the collected Web snippets. Text processing is required for extracting n-gram, because the n-gram, which is the synonym of the given *TL*, is usually outside of usual Chinese vocabulary [4]. The details of the text processing will be illustrated in Section 6.3.

The framework for confirming whether the pair is synonymous includes two major steps. First, we measure the phonetic similarity for the pair. This will be illustrated in Section 4. Second, we cast the confirmation problem as an ensemble task to make decision, using a majority-voting scheme and a boosting scheme together due to the fact that a pair must be learned more frequently when it is not easily confirmed. The decision-making process returns a result of either "synonymous" or "not synonymous" for the pair; then, the results are recorded in a knowledge database. It will be illustrated in Section 5.

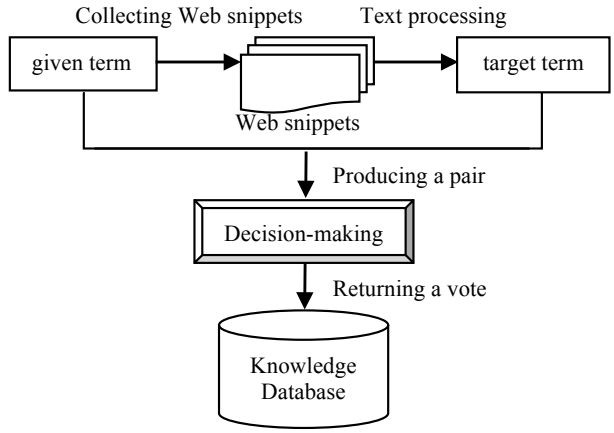


Figure 1. A framework for collecting pairs of synonymous transliterations from the Web

4 PHONETIC SIMILARITY BETWEEN CHINESE TRANSLITERATIONS

Judging similarity between Chinese transliterations is the main criterion for confirming a synonymous Chinese transliteration pair because the members in the pair are transliterated from a cognate foreign term, and so their phonetic similarity may be high. This criterion relies on three major processes (see Figure 2) to present how

to obtain the similarity score for a Chinese pair (a given term, a target term). The summary of the three processes is briefly discussed as follows.

- Assuming that we have a pair (i.e., a *TL* A and an n-gram B), a Romanization transcription system is used to transcribe the *TL* and the n-gram into two sets of sound alphabet sequences, where each Chinese character is transcribed into a sound alphabet sequence. For instance, a *TL* 梅爾吉普森 is transcribed into either a set of sound alphabet sequences (mei er ji pu sen) via the Pinyin system or a set of sound alphabet sequences (ㄇㄟˊㄦˇㄐㄧˊㄆㄨˇㄙㄣ) via the BPMF system. The details will be described in Section 4.1.
- To acquire the similarity score for the pair, a dynamic programming algorithm [26] is employed for calculating the maximum phonetic similarity score between two sets of sequences. The calculated similarity score is then normalized according to the length of the pair. The details will be described in Section 4.2.
- While running the dynamic programming algorithm, several pronunciation-based approaches are respectively used as the basis to assign similarity among sequences. The details will be described in Section 4.3.

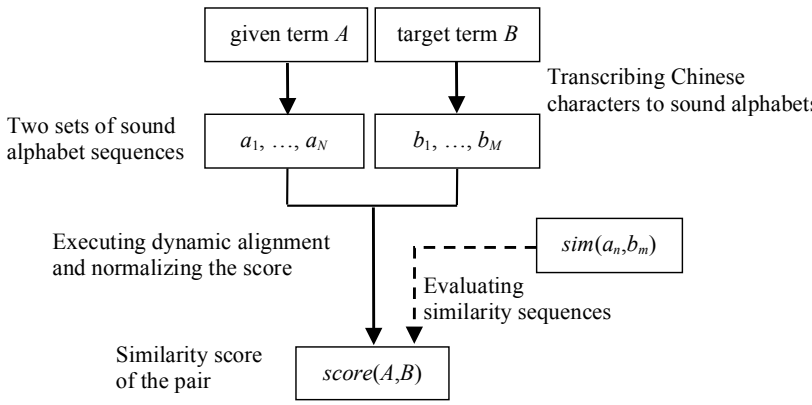


Figure 2. A procedure for calculating the similarity score for a Chinese pair

4.1 Phonetic Symbolic Representation for Chinese Characters

In the Mandarin Chinese linguistics, each Chinese character typically represents a syllable. A character could have more than one phonological value, and some characters could have the same phonological values. Two Romanization transcription systems, including the National Phonetic System of Taiwan (BPMF system used in Taiwan) and the Pinyin system (which is commonly used in China), are used to transcribe Chinese characters into sound alphabets.

In the BPMF system, there are 37 phonemes. There are 21 initial phonemes and 38 finals, including 16 base finals and 22 combinations of two phonemes. Each Chinese character is consisted of at most three phonemes which are initial, middle and final and has a tone. Ignoring tones, 412 symbolic compounds can represent all Chinese characters in pronunciation. In computational phonology, the pronunciation of Chinese characters provides a way to assign the similarity score between Chinese characters. Techniques such as CSC (character sound comparison method) [20] and LC (the study in Lin and Chen) [18] can be learned through the comparison of sequences using the BPMF system.

In the Pinyin system, a Chinese character is transcribed into grapheme-based sequence using Pinyin. ALINE (the system developed by Kondrak) [19], FSP (friction-strength pitch method) [16] and PLCS (pronunciation with longest common subsequence) [17] can be used for obtaining feature values for comparison between two sound alphabets.

4.2 Similarity Comparison for Chinese Transliterations

Measuring similarity for two sets of sound alphabet sequences produces a similarity score between two transliterations. This similarity is evaluated using the process of sequence alignment [27]. Assume that we have two Chinese transliterations $A = (a_1, \ldots, a_n, \ldots, a_N)$ and $B = (b_1, \ldots, b_m, \ldots, b_M)$, where a_n is the n^{th} character of A and b_m is the m^{th} character of B. N may not be equal to M. The characters a_n and b_m are formed into sound alphabet sequences $a_n = (a_{n,1}, \ldots, a_{n,i}, \ldots, a_{n,I})$ and $b_m = (b_{m,1}, \ldots, b_{m,j}, \ldots, b_{n,J})$, respectively. The alphabets $a_{n,i}$ and $b_{m,j}$ are generated by either the BPMF system or the Pinyin system.

To acquire the maximum similarity score between two sets of sound alphabet sequence (formed from A and B, respectively), which is represented as score(A, B), a dynamic programming-based approach can be used to acquire the largest distortion between A and B by adjusting the warp on the axis of $T(n, m)$ of $sim(a_n, b_m)$, which represents the similarity between a_n and b_m. The recursive formula (1) is defined as follows.

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

The base conditions are defined as $\{T(n,0)\}_{n=1}^{N} = 0$ and $\{T(0,m)\}_{m=1}^{M} = 0$. Figure 3 a) shows the running process of the comparison between a Chinese transliteration 梅爾吉普森 (ㄇㄟˊㄦˇㄐㄧˊㄆㄨˇㄙㄣ; mei er ji pu sen) and 梅爾吉勃遜(ㄇㄟˊㄦˇㄐㄧˊㄅㄛˊㄒㄩㄣ; mei er ji bo syun), whereas Figure 3 b) shows the running process for comparing the Chinese transliteration "艾布拉姆斯"(ㄞˋㄅㄨˋㄌㄚㄇㄨˇㄙ; ai bu la mu si) and "愛博斯"(ㄞˋㄅㄛˊㄙ; ai bo si). The anchor represents the dynamic path running the recursive formula. The bold connection path represents the alignment path and indicates that the maximum score between A and B is acquired by rendering the pair indicated by the bold arrow; thus, the anchor $T(N, M)$

is the finite termination of the recursive formula and results in a maximum similarity score.

$T(n,m)$	梅	爾	吉	勃	遜
梅	↖ **1**	← 1	← 1	← 1	← 1
爾	↑ 1	↖ **2**	← 2	← 2	← 2
吉	↑ 1	↑ 2	↖ **3**	← 3	← 3
普	↑ 1	↑ 2	↑ 3	↖ **3.77**	← 3.77
森	↑ 1	↑ 2	↑ 3	↑ 3.77	↖ **4.36**

a) (梅爾吉普森, 梅爾吉勃遜)

$T(n,m)$	愛	伯	斯
艾	↖ **1**	← 1	← 1
布	↑ 1	↖ **1.85**	← 1.85
拉	↑ 1	↑ **1.85**	↖ 2.28
姆	↑ 1	↑ **1.85**	↖ 2.28
斯	↑ 1	↑ 1.85	↖ **2.85**

b) (艾布拉姆斯, 愛伯斯)

Figure 3. Results of dynamic alignment for two chinese transliterations using a dynamic recursive formula. In this case, the sound alphabet uses BPMF system, and the evaluator of $sim(.,.)$ is CSC

To avoid longer transliterations that appear to be more similar and thus acquire higher $T(N,M)$, the similarity score must be normalized, taking into account the average length of the transliterations, as defined below.

$$\text{score}(A,B) = \frac{T(N,M)}{(N+M)/2}, \tag{2}$$

where the formula respects the similarity range $[0,1]$; accordingly, the two normalized scores in the above examples are 0.87 and 0.71, respectively.

4.3 Similarity Evaluation Among Sequences Based on Pronunciation Approaches

The BPMF system is used to transcribe a Chinese character into a phonetic sequence for the use of CSC and LC; the Pinyin system is used for ALINE, FSP and PLCS. Assume that we want to assign the similarity between two Chinese characters, 森 (sen) and 生 (sheng). Since the basis of the recursive formula (1) is $sim(a_n, b_m)$, CSC and LC assign $sim(\text{ㄙㄣ}, \text{ㄕㄥ})$, whereas the other techniques, including ALINE, FSP and PLCS, assign sim(sen, sheng). The assigned entries for $sim(.,.)$ are as follows.

4.3.1 CSC Approach

In previous studies on the signal processing, the similarity evaluator referred to as the Chinese character sound comparison (CSC) [20] was learned for the sound alphabet based on human pronunciation. The signal processing for pronunciation sound feature extraction can be divided into several major tasks, such as frame segmentation, endpoint detection and sound vector feature extraction with 26 dimensions, including 12 cepstral, 12 delta-cepstral coefficients, energy and delta-energy in the MFCC[1] domain. After this step, each sound was represented by a set of vectors of sound features. Therefore, after transcribeing the pronunciation sound into sound vectors, the sound similarity was used to construct two similarity matrices for Chinese character pronunciation, which includes a 37×37 phoneme sound matrix indicating the similarity scores among phonemes and a 412×412 syllable sound matrix indicating the similarities among syllables. These two matrices can represent the similarity in pronunciation between any two Chinese characters.

Consider the situation in which we want to acquire the similarity score $sim_{CSC}(.,.)$ using the CSC approach. According to our experience and that of Hsu et al. [20], the initial consonant of a Chinese character heavily influences the similarity of the pronunciation sound in comparison with others. Faced with this problem, we adopt an initial-weighted comparison approach involving a balancing adjustment; the similarity is weighted for the initial consonant of a phonetic sequence to balance the bias of a syllable. The equation $sim_{CSC}(a_n, b_m)$ is the weighted similarity score between two Chinese characters a_n and b_m with respect to the similarity matrices of the phonemes and the syllables.

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

where $s_{s37}(.,.)$ represents a function returning a phoneme similarity score, whereas $s_{s412}(.,.)$ represents a function returning a syllable similarity score. $a_n.IC$ and $b_m.IC$ represent the initial consonant (IC) for a_n and b_m, respectively. The parameter w facilitates the trade-off between an initial consonant and a syllable.

4.3.2 The LC Approach

An approach based on the scoring scheme in Lin and Chen [18] was designed for comparing a transliteration to its original English term. In this paper, we carefully adopt the scheme for a similarity comparison of Chinese characters. The similarity between phonemes is defined as follows. A matched consonant pair is assigned 10 points; otherwise, it is assigned -10 points. A matched vowel pair is assigned 5 points; otherwise, it is assigned 0. Matching with a null phoneme (or inserting a null phoneme) is assigned -5 points. As in Lin and Chen [18], to consider articulatory similarity, we assign some consonant pairs 8 points and vowel pairs

[1] MFCC – mel-frequency cepstrum coefficients

4 points. The pairs are determined by their articulatory features, which are identified based on their corresponding alphabets in the IPA. The consonant pairs include (ㄅ(b), ㄆ(p)), (ㄉ(d), ㄊ(t)), (ㄍ(g), ㄎ(t)), (ㄐ(j), ㄑ(q)), (ㄓ(zhi), ㄔ(chi)), (ㄗ(zi), ㄘ(ci)), (ㄕ(shi), ㄙ(si)), (ㄔ(chi), ㄘ(ci)), (ㄕ(shi), ㄖ(ri)) and (ㄓ(zhi), ㄗ(zi)). The vowel pairs include (ㄧ(yi), ㄩ(ü)), (ㄛ(o), ㄡ(ou)), (ㄢ(an), ㄤ(ang)) and (ㄣ(en), ㄥ(eng)). The maximum and the minimum similarity score of comparing two strings is obtained by $10 \times |c| + 5 \times |v|$ and $-10 \times (|c| + |v|)$, where $|c|$ and $|v|$ are the number of consonants and of vowels, respectively, in the longer string. The similarity range is normalized to the total matched score. For instance, sim(ㄍㄨㄛ(guo), ㄎㄡ(kou)) $= 8 - 5 + 4 = 7$, and the normalized score is 0.74 (i.e., $(7 - (-30))/(20 - (-30))$).

4.3.3 The ALINE Approach

This approach is based on the similarity scoring scheme of ALINE [19]. This scheme considers a set of operations, including insertions/deletions, substitutions and expansions/compressions. The phonetic features are used to score similarity between sound alphabets. In their default setting for the similarity assignment between two sound alphabets (represented as $s(a_{n,i}, b_{m,j})$), two identical consonants receive 35 points, while identical vowels receive 15. ALINE considers articulatory features and assigns different weights to the features according to their relative importance. Readers can refer to [19] for details. Normalization can be performed by dividing the final score by $35 \times |c| + 15 \times |v|$, where $|c|$ and $|v|$ are the numbers of consonants and vowels, respectively, in the longer string.

4.3.4 The FSP Approach

We use the feature scheme suggested in [16]. The scheme identifies two perceptual features or axes; namely friction strength (FS) and pitch (PI), and divides the consonant phonemes into six groups, differentiated by their score on each of these axes. For instance, bilabial plosive consonants (e.g., p, b) have a friction-strength score of 0 and a pitch score of 0, while alveolar fricative consonants (e.g., s, z) have scores 1.0 and 1.0, respectively. When comparing two vowels, we set the similarity score to 1 if they are identical; otherwise, it is set to 0. The similarity between two sound alphabets, as represented by $s(a_{n,i}, b_{m,j})$, uses the following equation.

$$s(a_{n,i}, b_{m,j}) = \begin{cases} 1, & \text{if } a_{n,i} = b_{m,j}, \\ 0.95, & \text{if } a_{n,i} \neq b_{m,j}, \text{ but} \\ & \text{in the same group,} \\ \frac{1.5-(|FS(a_{n,i})-FS(b_{m,j})|+0.5*|PI(a_{n,i})-PI(b_{m,j})|)}{1.5}, & \text{otherwise.} \end{cases} \tag{4}$$

4.3.5 The PLCS Approach

Assigning similarity score between two sequences can be induced to measure similar subsequences [28]. Gao et al. [13] used a dynamically discovered alignment to map an English phoneme sequence to a Romanized Chinese sound alphabet sequence. Tao et al. [14] utilized substitution, deletion and insertion operations in a dynamic programming application to determine the distance between two phonetic sequences. In this paper, we use the longest common subsequence (LCS) algorithm to find the longest common subsequence between two sequences. Two sequences are more similar if they have larger lengths of common subsequences. The different sound alphabets can be regarded as consistent in comparison, such as "p-b," "t-d", "k-g", and "n-ng", each pair of which has similar pronunciation places or pronunciation manners. Lin et al. [18] reported a set of corresponding pairs. Moreover, Chen et al. [17] mentioned that the first sound alphabet is more important than the others in a sequence; for example, "s" in "sen" is more important than "e" and "n" in "sen". We carefully study these aspects in the comparison of two phonetic sequences. Formula (5) is defined as

$$sim(a_n, b_m) = w \times \delta(f_{a_n}, f_{b_m}) + (1 - w) \times \frac{\delta(o_{a_n,b_m})}{\max(|o_{a_n}|, |o_{b_m}|)} \tag{5}$$

where w is a weighted trade-off parameter between two parts of the equation; $\delta(.,.)$ is a function that returns a total number of matched instances; f is the first letter of the sequence; and o is the subsequence, ignoring the first letter in a sequence.

5 DECISION-MAKING

We define a new ensemble scheme specified on the confirmation of Chinese transliteration pairs. Assume that we have a dataset X containing a set of pairs and that we want to confirm for each pair, which consists of a transliteration and an n-gram, whether the pairs are synonymous or not. We assume that we have a set of independent learning approaches M, each of which contributes a vote (or makes a decision) for each pair in X. $|M|$ votes are aggregated as the winning vote using a majority-voting scheme. In particular, the set of votes is obtained using CSC, ALINE, FSP, LC and PLCS.

The subsets of training data $X_1, X_2, \ldots, X_T$, which are generated by re-sampling a given training dataset, are learned as a set of classifiers $C_1, \ldots, C_t, \ldots, C_T$. An ensemble operation is then applied to obtain a winning vote from the results of the trained classifiers. The framework is briefly shown in Figure 4 and its detail is described as following.

5.1 Definition and Decision-Making Using a Similarity Entity

Let X be a dataset containing a set of n data pairs, and let $x_j \in X$ be a pair consisting of a transliteration and another Chinese term, which corresponds to class

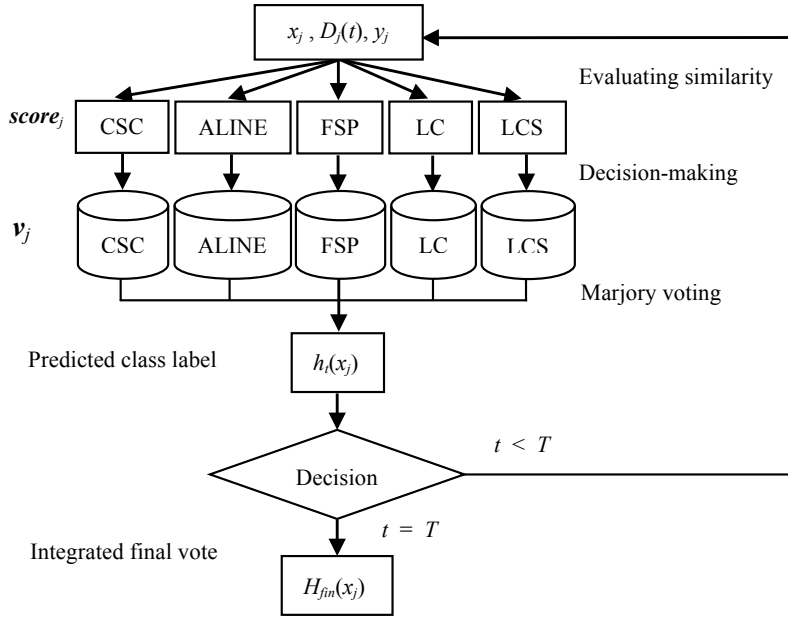


Figure 4. A framework for collecting pairs of synonymous transliterations

label $y_j \in \boldsymbol{Y}$, representing a synonymous pair or not a synonymous pair. Let $\boldsymbol{M} = [m_1, \ldots, m_I]^T$ be a set of pronunciation-based approaches, where m_i is the i^{th} approach in $\boldsymbol{M}$. For a pair $x_j \in X$, let $\boldsymbol{score}_j = [\text{score}_{j,1}, \ldots, \text{score}_{j,I}]^T$ be a set of similarity scores, where $\text{score}_{j,i}$ is measured by m_i (to use formula (2)) for x_j, and then let $\boldsymbol{v}_j = [v_j, 1, \ldots, v_{j,I}]^T$ be a set of decisions, where $v_{j,i}$ is a decision (i.e., a vote) taken from $\text{score}_{j,i}$. In particular, a pair x_j has three entities, namely, y_j, $\boldsymbol{v}_j$ and $\boldsymbol{score}_j$.

The similarity entity $\boldsymbol{score}_j$ drives the decision entity $\boldsymbol{v}_j$. Most studies in the literature often take a vote, represented as $v_{j,i}$ that is accepted when $score_{j,i} \geq \theta_i$, whereas vote $v_{j,i}$ is rejected when $score_{j,i} < \theta_i$. The parameter θ_i is a threshold. Determining a higher value for θ_i often brings higher precision but lower recall, whereas determining a lower value θ_i often brings lower precision but higher recall. Nevertheless, the determination of the appropriate parameters $\{\theta_i\}_{i=1}^I$ is usually empirical in many applications of information retrieval.

Instead of requiring the parameters $\{\theta_i\}_{i=1}^I$, we use the K-nearest neighbor algorithm to obtain $\boldsymbol{v}_i$ with the help of $\boldsymbol{score}_j$, because it provides a rule that x_j can be classified according to its K nearest neighbor pairs; by the same token, the vote $v_{j,i}$ is assigned by a majority vote on $\{v_{j\to k,i}\}_{k=1}^K$ with respect to $\{score_{j\to k,i}\}_{k=1}^K$, where "$j \to k$" represents the k^{th} nearest neighbor training pair of x_j. Initially, we denote $\{v_{r,i}\}_{i=1}^I = y_r$ in advance if x_r is a training pair.

Since a majority-voting scheme is a well-known integrated voting approach to generate a final decision, it is applied to obtain a class label. The class label y_j is determined using a majority-voting scheme on v_j. In particular, the voting function $h(x_j)$ determines a predicted class label via a majority vote of $\{v_{j,i}\}_{i=1}^{I}$ and is written as

$$h(x_j) = \arg\max_{y \in Y} \sum_{i=1}^{I} \delta(v_{j,i}) = y, \tag{6}$$

where the function δ returns a Boolean value.

5.2 Hypotheses Combination on the Confirmation of Chinese Transliteration Pairs

The ensemble framework proposed in this paper considers the use of multiple learning approaches $\boldsymbol{M}$ and subsets of training data $X_1, X_2, \ldots, X_T$. Let $\{m_i\}_{i=1}^{I}$ denote a set of learning approaches based on the pronunciation model and $\{X_t\}_{t=1}^{T}$ denote a set of training datasets generated by the use of a boosting scheme [23].

Following the generation of a subset, X_t is evolved from X_{t-1} using a bootstrapping strategy in the training process. It is worth mentioning that a pair must be learned more frequently when it is not easy to confirm. In other words, a pair x_j will appear much more possible in X_t while acquiring the wrong predication class label in X_{t-1}. In contrast, x_j, while contributing the correct class label in X_{t-1}, may not appear in X_t. The algorithm for confirmation is shown in Figure 5.

Figure 5 shows that the output of $H_{fin}(x_j)$, which is to integrate multiple votes of number T as a final vote, is performed while allowing T rounds. Thus, a T-dimensional voting vector is made for each x_j and given via $\{h_t(x_j)\}_{t=1}^{T}$. Additionally, a learning round acquiring an accuracy rate lower than the random guess accuracy (be $1-1/|y|$) will not contribute to the final vote. The final voting function H_{fin} for x_j is written as

$$H_{fin}(x_j) = \arg\max\,(y \in Y) \sum_{t|\epsilon_t \leq 0.5}^{T} w_t \times \delta(h_t(x_j) = y), \tag{7}$$

where h_t represents an integrated vote for $\{m_i\}_{i=1}^{I}$ at the t^{th} round, and the function δ returns a Boolean value. We extend $h(.)$, which was mentioned in formula (6), as a weighted majority-voting function $h_t(.)$ related to the various contributions of the set of approaches $\{m_i\}_{i=1}^{I}$ at the t^{th} round. In addition, the extended formula must take the parameter t into account. The extended equation is written as

$$h_t(x_j) = \arg\max_{y \in Y} \sum_{i=1}^{I} u_i^t \times \delta(v_{j,i}^t) = y. \tag{8}$$

Providing different voting confidences for a repeatable learning procedure is indeed necessary. In other words, it is quite understandable that $\{h_t(x_j)\}_{t=1}^{T}$ have

Input:
$\{x_j\}_{j=1}^n \in X$ with its class label $\{y_j\}_{j=1}^n \in \boldsymbol{Y}$ is given
the number of rounds T is assigned
the pronunciation-based approach $m_i \in \boldsymbol{M}$ is determined
$\boldsymbol{score}_j$ for i=1, …, n, which are calculated using (2)
let $\{D_j(t)\}_{j=1}^n = 1/n$ for t = 1

Do:
for t = 1, 2, …, T {
Bootstrapping a subset X_t with respect to $\{D_j(t)\}_{j=1}^n$
for $\{x_j\}_{j=1}^n$ {
for $\{m_i\}_{i=1}^I \in M$ {
get $v_{j,i}^t$
}
}
calculate $\{u_i^t\}_{i=1}^I$ in (10)
execute $h_t(x_j)$ using (8)
calculate w_t in (9)
update D_j(t+1) in (11)
}
Output:
execute $H_{fin}(x_j)$ using (7)

Figure 5. The confirmation algorithm for Chinese transliteration pairs

different weights with respect to their capabilities in their corresponding learning spaces $\{X_t\}_{t=1}^T$; in addition, the capabilities of comparison approaches $\{m_i\}_{i=1}^I$. Two weighted entities w_t and u_i^t are learned from the learning property for round t. We write

$$w_t = \frac{1 - \epsilon_t}{\epsilon_t}, \tag{9}$$

where ϵ_t is the probability of training error at the t^{th} round. In addition, we also write

$$u_i^t = \frac{1 - \tau_i^t}{\tau_i^t}, \tag{10}$$

where τ_i^t is the probability of training error in the comparison approach m_i at the t^{th} round.

The entities $\{D_j(t)\}_{j=1}^n$ are good candidates for driving the data distribution for X_t. The x_j obtaining the correct vote at round t will receive a lower probability value $D_j(t+1)$ and will be less likely to be drawn at round $t+1$. $D_j(t+1)$ is expressed as

$$D_j(t+1) = \begin{cases} D_j(t), & \text{if } h_t(x_j) \neq y_j, \\ D_j(t) \times \frac{\epsilon_t}{1-\epsilon_t}, & \text{else.} \end{cases} \tag{11}$$

6 EXPERIMENTS

The experimental datasets include a training dataset and a testing dataset based on a real application context. Confirming synonymous transliterations from a real-world Web corpus is valuable to support general search engines for retrieving complete search results because most transliterations are outside of regular Chinese dictionaries [4, 5]. Given a pair (a testing instance) while its class label is unknown, the training dataset is used in training to form a classifier in order to predict the class label for the pair. The source of the training dataset is from [20], and the testing dataset is generated from the Chinese-dominant Web pages collected by the Google search engine. In the following sections, we introduce the preparation of the experimental dataset and the process of the experiment. Then we analyze the experimental results.

6.1 Preparation of the Training Dataset

The data source is selected from the study in [20] in which the dataset contains 188 transliterations collected from Web news sources. These transliterations are proper names, including geographic, entertainment, sport, political and some personal names. They are built as a set of pairs, some of which are synonymous and others of which are not synonymous pairs. In other words, the class label of each pair is known in advance. The pairs are constructed as a training dataset and are used for decision-making.

In particular, a total number of 17 578 unique pairs (C_2^{188}) is obtained. However, we only allow the length disparity of the pair to be one because the size of difference between a Chinese transliteration and its synonym is usually no more than one [20]. For instance, the length difference of a synonymous pair (梅爾吉布生 (mei er ji bu sheng), 米路吉遜 (mi lu ji xun)) is one. From this point of view, many pairs can be ignored without allowing the length difference to exceed one; thus, we retain 12 006 pairs, which include 436 true-synonymous pairs and 11 570 pseudo-synonymous pairs (i.e., pairs that are not synonymous).

In order to reduce the likelihood of participative training data driving confirmation performance as well as to ignore the influences of an imbalanced training dataset, we perform a validation task involving ten different datasets selected from the training data by sampling without replacement and thus ensure the number of positive pairs is the same as the number of negative ones. Therefore, ten training datasets, each of which includes 436 positive pairs and 436 negative ones, are used for the experiments.

6.2 Description of the Input Transliterations

Two datasets, D50 and D97, are used for the experiments and contain transliterations. The D50 dataset includes 50 Chinese transliterations collected from the Web as shown in Table 1. Their lengths are 2, 3 or 4, which are the most commonly-seen

lengths in Chinese transliterations. The number of transliterations in each respective length group is 10, 30 and 10.

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

Table 1. Fifty original terms and their canonical transliterations

The D97 dataset is from the 2008 TIME 100 list of the world's most influential people [29]. There are 104 names in the list, since four entries include two names. Ninety-seven names are retained for the experiment. Seven names are ignored, namely, Ying-Jeou Ma, Jintao Hu, Jeff Han, Jiwei Lou, Dalai Lama, Takahashi Murakami, and Radiohead. The first five have Chinese last names that have standard Chinese translations. The sixth term is a Japanese name for which translation is usually not done using transliteration. The last name is that of a music band; its translation to Chinese is not according to its pronunciation, but its meaning.

6.3 Constructing Pairs from the Web

In this experiment, we input the transliterations in D50 and D97 to collect their synonyms from a real-world Web corpus using the integrated confirmation framework proposed in this paper. For each transliteration, we collected Web snippets

by submitting a search keyword to the Google search engine. The search keyword is used to retrieve Web snippets; however, it does not contribute information to the confirmation framework, which determines whether a pair is synonymous.

To construct a pair, we use the original term of the given transliteration as a search keyword, because the original term is able to retrieve appropriate Web documents in which the transliteration's synonyms appear. Let a transliteration (abbreviated as *TL*) be an entry. The *TL*'s original term (abbreviated as *ORI*), which is treated as the search keyword for the search engine, is represented as Q_{Ori} and is submitted to retrieve search result Web snippets, represented as D_{Ori}. The set D_{Ori} is limited to Chinese-dominant Web snippets. The procedure of returning a pair by collecting Web snippets from the Google search engine is as follows.

A. For each *TL* in D50 and D97, we use Q_{ORI} to download Web snippets D_{ORI}. In particular, we set $|D_{ORI}|$ to 20 for each *TL* because the snippets appearing at the head of the returned snippets are often more relevant to the research keyword. The size of the downloaded D_{ORI} for D50 is 1 000, whereas the size of the downloaded D_{ORI} for D97 is 1 940.

B. We delete known vocabulary terms with the help of a Chinese dictionary for D_{ORI} and apply an N-gram algorithm to segment Chinese n-grams for the remaining fractional sentences in D_{ORI}. Furthermore, most synonymous transliterations (*TLs* with their *STs*) have the same length, but some of them have different lengths of at most one [20]. Therefore, we retain the Chinese terms from D_{ORI} while controlling for length. Each Chinese term of length N is retained, with $N = |TL| - 1$ to $N = |TL| + 1$ and $N \geq 2$. The number of remaining pairs for D50 is 9 439, whereas that for D97 is 19 263, where the pair consists of a given TL and a remaining Chinese n-gram.

C. However, some pairs have similarities that are not high enough and thus are never considered synonymous pairs. We set a similarity threshold to ignore those pairs. According to the findings in [20], a lower similarity threshold θ can be set to 0.5 by using the CSC approach to cover effectively all examples of synonymous transliterations. After discarding the pairs with similarities lower than 0.5, 2 132 and 5 324 pairs are retained for D50 and D97, respectively. These pairs are confirmed by the use of the framework proposed in this paper and will be discussed in next section.

6.4 Confirmation of Synonymous Transliterations and Performance Analysis

The experiments demonstrate whether the proposed framework is effective in extracting synonymous transliterations from the Web. The following nine approaches are employed for comparison in the experiments.

- The integrated confirmation framework (ICF): This is the ensemble framework proposed in this paper.

- The majority-voting approach (MV): This is a simple ensemble approach. Each classifier contributes an equal vote. In particular, we use Equation (6) to perform this approach.
- The individual approach: There are five approaches, CSC, LC, ALINE, FSP and PLCS, each of which is performed individually in the experiment.

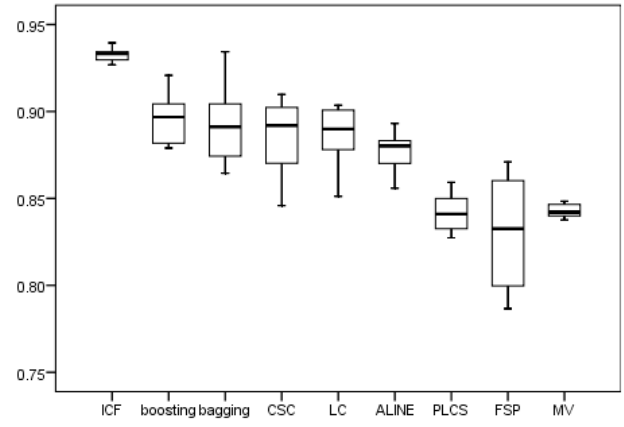
A feature vector with five dimensions generated using these five approaches can be performed for the experiments; hence, a classification-learning algorithm can be applied to predict the class label for each 5-tuple pair. The following two approaches are popular for analyzing experiments in the literature and are employed for comparison in this paper.

- Bagging: This combines multiple classifiers to predict the class label for a pair by integrating their corresponding votes. The base algorithm for the classification we used is *KNN* with $k = 5$ due to its simplicity.
- Boosting: This requires a weak learning algorithm. We use *KNN* with $k = 5$ in this study.

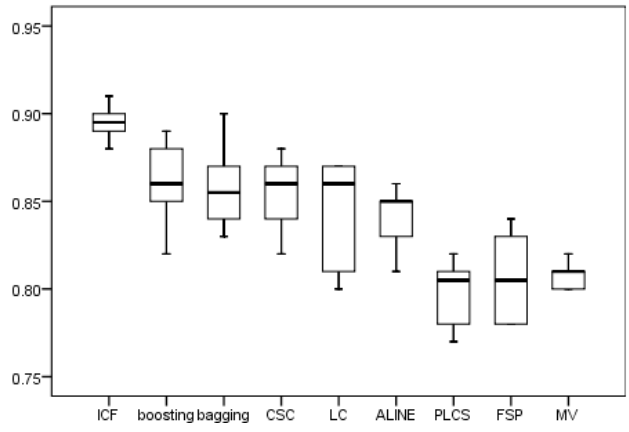
ICF, bagging and boosting are the same in that they determine a parameter T, the number of iterations. One study [23] set the parameter T to 10 to use the boosting scheme. We follow the same setting for our experiments. Therefore, ten results are obtained for the testing data in the experiment, since we have ten training datasets involved in the validation process. The evaluator used for the experiment is the accuracy measure, which is common in a classification task. Moreover, we use a box-plot analysis to graphically employ nine approaches, including ICF, boosting, bagging, MV, and five individual approaches (CSC, LC, ALINE, FSP and PLCS). The results are shown in Figure 6.

In Figure 6, the experimental results show that the average accuracy in the confirmation of Chinese transliteration pairs for three ensemble approaches (namely, ICF, boosting, and bagging) is higher than that of the other individual approaches. In addition, ICF achieves an average accuracy of 0.93 in D50 and 0.89 in D97 and is the best among the nine approaches. CSC achieves an average accuracy of 0.88 in D50 and 0.85 in D97 and is the best among the five individual approaches. Moreover, a shorter distance between the top and the bottom in a box-plot analysis demonstrates that ICF produces a much more stable performance than the others do; in contrast, bagging produces the most unstable performance among all ensemble approaches. All five individual approaches produce a less stable performance than the ensemble approaches, because they are seriously affected by the training datasets.

We give solid analyses for boosting and bagging as compared to our ICF due to their performance improvement. The ICF, boosting and bagging methods are the same in that they create multiple classifiers. As such, each classifier contributes a vote whose weight can be designed using comprehensive strategies. In particular, we use four strategies for ICF, boosting and bagging. For the details of these



a)



b)

Figure 6. Box-plot analysis for nine approaches in the testing datasets a) D50 and b) D97

strategies, please refer to [2]. The first strategy is simple voting and the other three strategies are weighted voting, which requires tuning the weights on training data. These strategies are shown as follows.

- Equal-Voting: Each classifier contributes an equal vote.
- Precision-Weight: The weight is measured by precision measure.
- classPrecision: The weight is measured by class precision; thus, a classifier can prefer various weights for predicting different class labels.
- Precision-Recall: The weight is measured by two measures which are precision and recall.

We use these four strategies for ICF, boosting and bagging on the D50 and D97 datasets. We perform each experiment ten times and the performance is evaluated by an accuracy evaluator. Table 2 and Table 3 show the average accuracy on D50 and D97 datasets, respectively.

From the overall data in Table 2 and Table 3, ICF has better identification performance than boosting and bagging when using the four strategies, and bagging usually results in bad performance. Furthermore, we then observe the performance of using four strategies for ICF, boosting and bagging. In Table 2, we can see that the classPrecision strategy is very helpful to ICF and boosting, and the Precision-Recall strategy is helpful to bagging. In Table 3, we see that the Precision-Recall strategy is adaptively used for ICF, boosting and bagging. However, the Equal-Voting strategy results in worse performance.

	ICF	boosting	bagging
Equal-Voting	87.9 %	87.3 %	86.8 %
Precision-Weight	92.3 %	89.4 %	89.0 %
classPrecision	93.1 %	90.5 %	88.6 %
Precision-Recall	92.7 %	88.6 %	89.8 %

Table 2. Average accuracy on three ensemble schemes with four strategies on D50 dataset

	ICF	boosting	bagging
Equal-Voting	85.4 %	85.9 %	84.2 %
Precision-Weight	89.3 %	86.3 %	86.8 %
classPrecision	89.9 %	86.7 %	85.5 %
Precision-Recall	90.7 %	88.2 %	87.3 %

Table 3. Average accuracy on three ensemble schemes with four strategies on D97 dataset

According to the results of the individual approaches, we find that not all approaches should be used in the three ensemble approaches (that is, ICF, boosting, and bagging). This seems to address the issue of whether fewer approaches are enough to achieve ensemble voting. Therefore, the top r ranked approaches, ordered as CSC, LC, ALINE, PLCS and FSP, are considered the candidate participant procedures; the top ranked approach brings better performance. This order has resulted from the tuning of every individual approach on the training dataset, where we use average accuracy as a measure to evaluate the tuning performance. Therefore, we build three extended approaches, namely, ICF_r, boosting_r and bagging_r, derived respectively from ICF, bagging and boosting. The parameter r is set to 2 and 3, respectively. For example, ICF_2 indicates that we use two top-ranked approaches (CSC, LC), whereas ICF_3 indicates that we used three top-ranked approaches (CSC, LC, ALINE). Figure 7 and Figure 8 show the experimental results for D50 and D97 datasets, respectively.

From the results in Figure 7 and Figure 8, the performance in terms of classification accuracy rate and stability for ICF_r is better than that for either boosting_r or

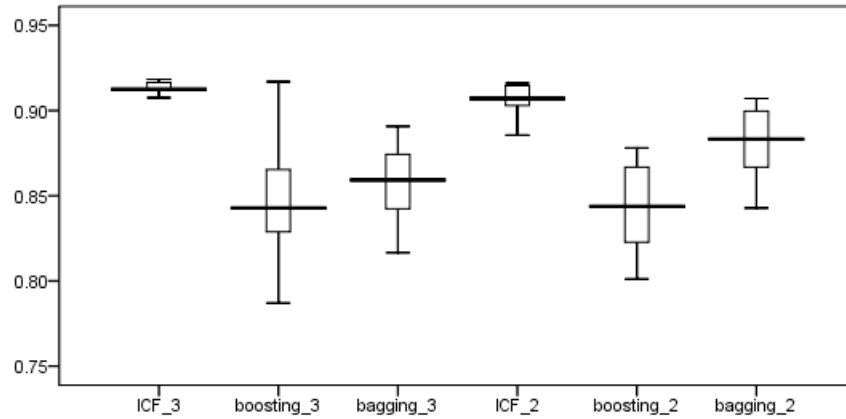


Figure 7. Box-plot analysis for three ensemble approaches using r top-ranked approaches on D50 dataset

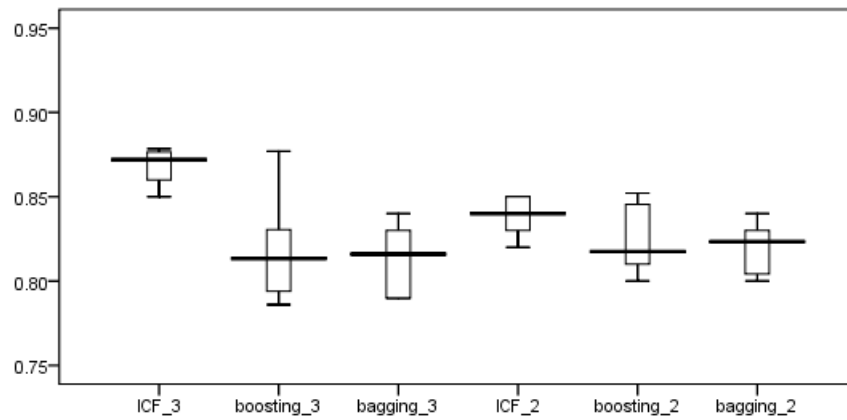


Figure 8. Box-plot analysis for three ensemble approaches using r top-ranked approaches on D97 dataset

bagging_r. It can be inferred that the participating approaches in ICF_r contribute different weights with respect to the use of Equation (10). The use of fewer approaches in the ensemble approach is potentially desirable, but this achieves worse performances.

The notions of type I and type II errors are well accepted to analyze these performances and can be understood as the false positive pairs and the false negative pairs, respectively, in a confusion matrix. We illustrate the results of the confusion matrices generated from the ICF approach in the two datasets D50 and D97 and observe the differences at t rounds as 1 and 10, respectively. The results of the

confusion matrices are respectively shown in Figure 9 and Figure 10, where the values are averaged from 10 tests.

		Actual condition	
		Actual-ST	Pseudo-ST
Test result	Actual-ST	28.8	241.5
	Pseudo-ST	9.2	1852.5

a) $t = 1$

		Actual condition	
		Actual-ST	Pseudo-ST
Test result	Actual-ST	33.3	139.7
	Pseudo-ST	4.7	1 954.3

b) $t = 10$

Figure 9. Confusion matrices for the ICF approach for D50 dataset

		Actual condition	
		Actual-ST	Pseudo-ST
Test result	Actual-ST	20.4	956.3
	Pseudo-ST	21.6	4 325.7

a) $t = 1$

		Actual condition	
		Actual-ST	Pseudo-ST
Test result	Actual-ST	34.2	547.8
	Pseudo-ST	7.7	4 734.2

b) $t = 10$

Figure 10. Confusion matrices for the ICF approach for D97 dataset

The performance in terms of classification accuracy rate has increased. We can see that the average number of misclassified examples is 250.7, of which the average number of false positive examples (i.e., type I errors) is 241.5 and that of false negative examples (i.e., type II errors) is 9.2 when we set t to 1 as shown in Figure 9 a). In comparison, when we set t to 10 as shown in Figure 9 b), the average number of misclassified examples is 144.4, of which the average number of false positive is 139.7 and that of false negative is 4.7. Figure 10 a) and Figure 10 b) show similar trends.

We briefly summarize the retrieved synonymous transliterations from the dataset D50 in order to test the ICF approach. The pairs that are often true positive are shown in Table 4, and those that are often false positive are shown in Table 5. This also indicates that more than one synonymous transliteration is potentially extracted for several given transliterations.

TL	*Ori*	Extracted *ST*	*TL*	*Ori*	Extracted *ST*
喬登	Jordan	喬丹	迪士尼	Disney	迪斯尼
詹森	Jansen	楊森	柴契爾	Thatcher	撒切爾
蓋亞	Gaea	嘎雅	傑克遜	Jackson	杰克遜
蓋茲	Gates	蓋斯, 蓋茨	歐尼爾	O Neal	奧尼爾
鮑爾	Powell	鮑威, 鮑威爾, 保威	歐萊禮	Reilly	奧萊理
巴非特	Buffett	巴菲特, 比菲特	篷比杜	Pompidou	龐比度, 蓬皮杜
巴薩拉	Basra	巴斯拉, 彼特拉	羅伯茲	Roberts	羅伯茨
比卡丘	Pikachu	皮卡丘, 皮卡秋 比卡秋	戈巴契夫	Gorbachev	哥巴契夫
加奈特	Garnett	加內特	史瓦辛格	Schwarzenegger	施瓦辛格
皮爾遜	Pearson	皮爾森	史柯西斯	Scorsese	斯科塞斯, 史高西斯
貝克漢	Beckham	貝克漢姆	奈西利亞	Nasiriyah	納西里耶
所羅門	Solomon	索羅門			

Table 4. True positive pairs in testing ICF for D50

TL	*Ori*	miss-classified *ST*
布希	Bush	布什
喬登	Jordan	佐丹
雪梨	Sydney	悉尼
胡笙	Hussein	侯賽因
華勒沙	Walesa	瓦文薩

Table 5. False negative pairs in testing ICF for D50

7 CONCLUSION AND FUTURE WORK

In this paper, we propose a new ensemble framework for confirming Chinese transliteration pairs. Our framework confirms and extracts pairs of synonymous transliteration from a real-world Web corpus, which is helpful to support search engines such as Google and Yahoo for retrieving complete search results. The framework for confirming Chinese transliteration pairs includes two steps. First, we study the Romanization transcription systems, including the BPMF system and the Pinyin system, to describe Chinese characters into sound alphabets, and then we adopt five pronunciation-based approaches to measure the similarity between a Chinese transliteration and another Chinese term. Second, our framework considers the use of the majority-voting scheme and the boosting scheme at the same time. The experimental results were evaluated according to the proposed framework in this paper, comparing boosting [1], bagging [3] and five individual approaches. In addition, we use simple voting and several weighted voting for ICF, boosting and bagging. The experimental results demonstrate that the proposed framework is robust for improving performance in terms of classification accuracy and stability.

Transliteration comparison is a core step in many practical applications, including topic detection, tracking of news and information retrieval via search engines. Collecting the pairs of synonymous Chinese transliteration correctly is crucial for the real-world applications. We plan to pursue some of these issues in the future based on these research results.

Acknowledgment

This research was supported by the National Science Council, Taiwan under Grant NSC 99-2410-H-146-001-MY2.

REFERENCES

[1] Freund, Y.—Schapire, R. E.: Experiments with a New Boosting Algorithm. Proceedings of the 13th International Conference on Machine Learning, 1996, pp. 148–156.

[2] van Halteren, H.—Zavrel, J.—Daelemans, W.: Improving Accuracy in Word Class Tagging Through Combination of Machine Learning Systems. Computational Linguistics, Vol. 27, 2001, No. 2, pp. 199–230.

[3] Breiman, L.: Bagging Predictors. Machine Learning, Vol. 24, 1996, pp. 123–140.

[4] Kuo, J.-S.—Li, H.—Yang, Y.-K.: A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs. ACM Transactions on Asian Language Information Processing, Vol. 6, 2007.

[5] Oh, J.-H.—Isahara, H.: Mining the Web for Transliteration Lexicons: Joint-Validation Approach. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 254–261.

[6] Oh, J.-H.—Choi, K.-S.—Isahara, H.: A Machine Transliteration Model Based on Correspondence Between Graphemes and Phonemes. ACM Transactions on Asian Language Information Processing, Vol. 5, September 2006, pp. 185–208.

[7] Knight, K.—Graehl, J.: Machine Transliteration. Computational Linguistics, Vol. 24, 1998, pp. 599–612.

[8] AbdulJaleel, N.—Larkey, L. S.: Statistical Transliteration for English-Arabic Cross Language Information Retrieval. Proceedings of ACM Conference on Information and Knowledge Management, 2003.

[9] Virga, P.—Khudanpur, S.: Transliteration of Proper Names in Crosslingual Information Retrieval. Proceedings of ACL 2003 Workshop on Multilingual and Mixed-Language Named Entity Recognition, 2003, pp. 57–64.

[10] Li, H.—Zhang, M.—Su, J.: A Joint Source-Channel Model for Machine Transliteration. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004, pp. 159–166.

[11] Wan, S.—Verspoor, C. M.: Automatic English-Chinese Name Transliteration for Development of Multilingual Resources. Proceedings of 17th COLING and 36th ACL, Montreal, Quebec, Canada, 1998, pp. 1352–1356.

[12] Oh, J.-H.—Choi, K.-S.: An Ensemble of Transliteration Models for Information Retrieval. Information Processing and Management, Vol. 42, July 2006, pp. 980–1002.

[13] Gao, W.—Wong, K.-F.—Lam, W.: Phoneme-Based Transliteration of Foreign Names for OOV Problem. Proceedings of the 1st International Joint Conference on Natural Language Processing, Sanya, Hainan, China, 2004, pp. 374–381.

[14] Tao, T.—Yoon, S.-Y.—Fister, A.—Sproat, R.—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, July 22–23, 2006.

[15] Simon, P.—Huang, C.-R.—Hsieh, S.-K.—Hong, J.-F.: Transliterated Named Entity Recognition Based on Chinese Word Sketch. International Journal of Computer Processing of Oriental Languages, Vol. 21, 2008, pp. 19–30.

[16] Connolly, J. H.: Quantifying Target-Realization Differences. Clinical Linguistics & Phonetics, Vol. 11, 1997, pp. 267–298.

[17] Chen, H. H.—Huang, S. J.—Ding, Y. W.—Tsai, S. C.: Proper Name Translation in Cross-Language Information Retrieval. Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, Montreal, Quebec, Canada, 1998.

[18] Lin, W. H.—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.

[19] Kondrak, G.: Phonetic Alignment and Similarity. Computers and the Humanities, Vol. 37, 2003, pp. 273–291.

[20] Hsu, C. C.—Chen, C. H.—Shih, T. T.—Chen, C. K.: Measuring Similarity Between Transliterations Against Noise Data. ACM Transactions on Asian Language Information Processing, Vol. 6, 2007, pp. 1–20.

[21] Kuncheva, L. I.: A Theoretical Study on Six Classifier Fusion Strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 2002, pp. 281–286.

[22] Kuncheva, L. I.—Whitaker, C. J.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Machine Learning, Vol. 51, 2003, pp. 181–207.

[23] Sun, Y.—Wang, Y.—Wong, A. K. C.: Boosting an Associative Classifier. IEEE Transactions on Knowledge and Data Engineering, Vol. 18, 2006, pp. 988–992.

[24] Dietterich, T. G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning, Vol. 40, 2000, No. 2, pp. 139–157.

[25] Kuo, J. S.—Li, H.—Yang, Y. K.: A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs. ACM Transactions on Asian Language Information Processing, Vol. 6, 2007.

[26] Cormen, T. H.—Leiserson, C. E.—Rivest, R. L.: Introduction to Algorithms: The MIT Press, 1996.

[27] Lopes, H. S.—Lima, C. R. E.—Moritz, G. L.: A Parallel Algorithm for Large-Scale Multiple Sequence Alignment. Computing and Informatics, Vol. 29, 2010, No. 6+, pp. 1233–1250.

[28] Deorowicz, S.—Obstój, J.: Constrained Longest Common Subsequence Computing Algorithms in Practice. Computing and Informatics, Vol. 29, 2010, No. 3, pp. 427–445.

[29] Time: The 2008 Time 100: The World's Most Influential People. `http://www.time.com/time/specials/2007/0,28757,1733748,00.html?iid=redirect-time100` (Retrieved on Feb. 5, 2009), 2009.

Chien-Hsing Chen received his Ph.D. degree in information management from the National Yunlin University of Science and Technology, Taiwan, in 2010. He is currently Assistant Professor at the Department of Information Management, Ling Tung University in Taiwan. His principal research interests are the area of information retrieval, machine transliteration, feature selection and data mining.

Chung-Chian Hsu received his M.Sc. and Ph.D. degrees in computer science from the Northwestern University, Evanston, IL, in 1988 and 1992, respectively. He joined the Department of Information Management, National Yunlin University of Science and Technology, Douliou, Taiwan, in 1993. He is currently Professor with the Department. He has been the Director of the Information Office, Testing Center for Technological and Vocational Education Taipei, Taiwan, since 2002. His current research interests include data mining, machine learning, pattern recognition, information retrieval, and decision support systems.