Synonymous Chinese Transliterations Retrieval from World Wide Web by Using Association Words

Chung-Chian Hsu and Chien-Hsing Chen

National Yunlin University of Science and Technology, Taiwan
{hsucc,g9423809}@yuntech.edu.tw

Abstract. We present a framework for mining synonymous transliterations from a set of Web pages collected via a search engine. An integrated statistical measure is proposed to form search keywords for a search engine in order to retrieve relevant Web snippets. We employ a scheme of comparing the similarity between two transliterations to aid in identifying synonymous transliterations. Experimental results show that the average number of harvesting synonymous transliterations is about 5.04 for an input transliteration. The retrieval results could be beneficial for constructing ontology, especially, in the domain of foreign person names.

Keywords: synonymous transliteration, cross lingual information retrieval, Chinese transliteration, person names, ontology.

1 Introduction

A *transliteration* is a local representation of a foreign word by rendering the pronunciation in the alphabet to the local language. With many different translators working without a common standard, there may be many different transliterations for the same proper noun. For example, the inconsistent Chinese transliterations 賓拉登 (bin la deng), 本拉登 (ben la deng) and 本拉丹 (ben la dan) are all translated from a foreign name "Bin Laden". Unfortunately, a person may know only one of those transliterations. As a result, the synonymous transliterations problem may engender comprehensive obstacle while one is reading. More importantly, it also results in incomplete search results when a user inputs only one of the transliterations to a search engine. For instance, using 賓拉登 (bin la deng) as a search keyword cannot retrieve the Web pages which use 本拉登 (ben la deng) as the transliteration for Bin Laden. In this paper, we attempt to propose a framework for automatically extracting as many synonymous transliterations as possible from the Web with respect to a given input transliteration as a first step to the problem. The research result is beneficial to constructing ontology, especially, in the domain of famous person names.

Some major tasks in natural language processing such as machine translation, named entity recognition, automatic summarization, information extraction and cross-language information retrieval (CLIR) have treated Web corpora as a good knowledge

M. Bubak et al. (Eds.): ICCS 2008, Part I, LNCS 5101, pp. 913–922, 2008.

source for extracting useful information. Search engines have been considered an important tool to retrieve relevant documents. However, a simple, short query usually fail in returning only highly relevant documents and instead a huge amount of Web pages in diversified topics are usually returned. A short query expanded by additional relevant search keywords could help to limit the retrieved pages to what the user is intended. Work in the literature such as query extension [1] proposed some techniques for identifying proper keywords for extension. We follow this idea for collecting high quality candidate snippets which might contain synonymous transliterations.

The traditional approaches in CLIR usually require a parallel corpus which suffers from bias and time-consuming due to manually collecting. Instead, we propose an effective framework to mining synonymous transliterations from Web snippets returned by a search engine. A critical step is to use proper keywords for collecting a limited amount of snippets which could include as many synonymous transliterations as possible. To achieve this goal, we use a measure which integrates several statistic approaches of keyword determination so as to raise the keyword quality. After retrieving relevant documents via a search engine, we apply a comparison scheme to determine whether an unknown word segmented from the retrieved snippets is indeed a synonymous transliteration. Our scheme is based on comparing digitalized physical sounds of Chinese characters. The traditional approaches in CLIR are usually grapheme-based or phonetic-based. Compared to those approaches, our approach possesses more powerful discrimination capability.

2 Candidate Snippets Collection

We propose a procedure as presented in Fig. 1 for collecting candidate Web snippets in which synonymous transliterations may appear. First, the transliteration (*TL*) is inputted for collecting a set of *n* snippets, called core snippets. After text preprocess, a set of *m* keywords, called association words, which are highly associated with the TL are extracted from the core snippets. The associated words are to form search-keywords to retrieve a set of *k* snippets from the Web, called candidate snippets, which are considered likely containing synonymous transliterations.

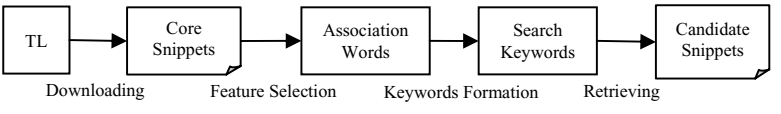

Fig. 1. A procedure of collecting candidate Web snippets

2.1 Association Words Selection

Several statistical methods [2] can be used to select feature terms with respect to a document category by measuring association strength between a term and the category, including Information gain (IG), Mutual information (MI), Chi-square (CHI), Correlation coefficient (CC), Relevance score (RS), Odds ratio (OR) and GSS Coefficient (GSS). A fusion approach which integrates features selected by different methods may improve the quality of features, reduce noisy, and avoid overfitting [2].

Therefore, to estimate the strength of association between a term t_k and an input transliterations c_i, we employ a fusion model integrating six popular feature selection functions.

To calculate the strength, we need to compute various joint and conditional probabilities. Recently, several researchers proposed to use the returned count of a query to a search engine for estimating term relationship. Cheng et. al. [3] used the returned page counts from the search engine to estimate association strength between two terms. Cilibrasi and Vitanyi [4] used the returned page counts to measure the information distance so as to estimate the similarity among the names of objects. We follow their idea for our needs. To take GSS as an example, $GSS(t_k,c_i) = p(t_k,c_i)p(\overline{t_k},\overline{c_i}) - p(t_k,\overline{c_i})p(\overline{t_k},c_i)$ where $p(t_k,c_i)$ represents the probability of co-occurrence of t_k and c_i which can be estimated via the returned page counts of a query "t_k" + "c_i" to a search engine, in which c_i is a transliteration and t_k is a term. t_k and c_i represent the positive existence of in a Web page containing while and indicate the opposite, non-existence.

In practice, we first download a fixed number of Web snippets D for a transliteration c_i via a search engine. Denote $T = \{t_1, \ldots, t_k, \ldots, t_K\}$ be a set of terms obtained from the core snippets, all terms $\{t_k\}_{k=1}^{K}$ in D are extracted and the scores on the six functions for association strength between t_k and c_i are measured. Six ranking values $\{r_k^m\}_{m=1}^{6}$ for each t_k with respect to the six functions are obtained, where represents the rank of t_k under the m^{th} evaluation function. The average rank f_k is defined as $f_k = \sum_{m=1,M} r_k^m/M$. A lower rank represents more important of the term.

2.2 Search-Keywords Formation

Based on the ranked association words selected in the previous step, there are several alternatives to form a query for further collecting candidate snippets which may contain synonymous transliterations. We consider several strategies and empirically compare their performance. Three entities are used to form different strategies for synonymous transliterations (*ST*), which are the transliteration *TL*, the association term (*AS*), and *TL*'s original word (*ORI*). Three strategies were made as follows.

Strategy 1 (Direct strategy). An *ST* may appear in the same snippet with a *TL* or *ORI*. Therefore, the *TL* or *ORI* can be used as the query term. Given a transliteration, its foreign origin can be determined automatically by several techniques found in CLIR [5-12].

Strategy 2 (Indirect strategy). Association words highly related to the *TL* are possible to retrieve snippets containing an *ST*. Therefore, in the indirect strategy, we make a query Q out of association words; specifically, a query $Q_{m\text{-}As}$ is an m-term query which is formed by m association words. We select significant association words and use a combination to generate a set of queries Q. Then, each of the queries is used to collect several hundred snippets which collectively form the set of candidate Web snippets. For instance, given the top four association words of 賓拉登 (Bin Laden), say {恐怖分子 (terrorist), 阿富汗 (Afghanistan), 攻擊 (attack), 恐怖主義 (terrorism)}, and $m = 2$, a query $Q_{2\text{-}As}$ is a 2-term query such as (恐怖份子,阿富汗).

The query set Q consists of all two-term combinations of the four *AS*s. The size of Q is $C(4,2) = 6$. The set of search-keywords in query $Q_{2\text{-}As}$ is {q_1= (恐怖份子,阿富汗); q_2=(阿富汗攻擊); q_3=(恐怖主義,攻擊); …; q_6}.

Strategy 3 (Integrated strategy). A combination of the direct and the indirect strategy may improve retrieval effectiveness. Therefore, an integrated strategy containing the $Q_{m\text{-}As}$ and the Q_{ORI} or Q_{TL} is considered. Empirically, the integration with *ORI* is much better than *TL*. Thus, we integrate association words with the *ORI* to produce a query $Q_{m\text{-}AsOri}$. For example, $Q_{1\text{-}AsOri}$ =(恐怖份子, Bin Laden) or $Q_{2\text{-}AsOri}$ =(恐怖主義、阿富汗、Bin Laden).

3 Synonymous Transliterations Extraction from Candidate Snippets

After collecting candidate snippets from the Web, we apply several processes to extract synonymous transliterations. Transliterations are unknown to an ordinary dictionary, so we first discard known words in the snippets with the help of a dictionary and then extract *n*-gram terms from the remaining text. The length parameter *n* is set to the range from |*TL*| - 1 to |*TL*| + 1 since the length of an *ST* with respect to an input *TL* is most likely in that range. A process of dynamic alignment is employed to select candidate synonymous transliterations (*CSTs*) from the *n*-gram terms. Then, we compare the similarity between *CSTs* and the *TL*. A highly similar *CST* to the *TL* is considered a synonymous transliteration. The extraction procedure is presented in Fig. 2.

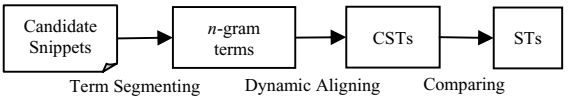

Fig. 2. The procedure of extracting synonymous transliterations

3.1 N-Gram Terms and Candidate Synonymous Transliterations Generations

The size of *n*-gram terms segmented from the remaining text after discarding known words is still huge. Most of them are obviously not an *ST*. We apply a heuristic to discard those *n*-gram terms and the remaining terms are referred to as candidate synonymous transliterations (*CSTs*).

In particular, we observed that most of synonymous transliterations are highly matching in the first and the last Character, for instance, 戈巴契夫 (ge ba qi fu), 戈爾巴喬夫 (ge er ba qiao fu) and 戈巴卓夫 (ge ba zhuo fu). That is to say, two terms which does not match well in the first and the last character are very likely not synonymous, for instance, 本拉丹 (ben la dan) and 拉丹襲 (la dan xi) in which the

first character of the latter matches with the second character of the former while the last character of the former matches with the second of the latter. In fact, 拉丹襲 which has the first two characters from a true synonymous transliteration 本拉丹 is generated due to the use of 3-gram segmentation.

An *extra-last-character exception* has to be taken care. Several final foreign phonemes might be ignored in the transliterations by some translators but not be ignored by some other translators. Those phonemes include "m", "er", "d" and "s" and usually be transliterated as 姆 (mu), 兒 (er), 爾 (er), 德 (de), and 斯 (si) when they are not ignored. For example, 貝克漢 (bei ke han) and 貝克漢姆 (bei ke han mu). Therefore, when a mismatched last character pair is attributed to this exception, we need to further explore the matching between the last second character of the longer word and the last character of the shorter word.

According to the above observations, we resort to a dynamic programming technique [12, 13] to determine the optimal alignment between an *n*-gram term and the *TL* so as to eliminate the *n*-gram terms which do not match well with the *TL* in the first and the last character and neither fall in the extra-last-character exception. Those *n*-gram terms which match well with the *TL* in the first and the last character or are well handled by the extra-last-character exception are considered *CSTs*. Note that the alignment is based on pronunciation similarity of Chinese characters.

3.2 Candidate Synonymous Transliterations Comparison

A transliteration usually has pronunciation close to their original foreign words. Therefore, synonymous transliterations usually have similar pronunciations. We use the Chinese Sound Comparison method (CSC) [12] to compare the pronunciation of two Chinese words, which has advantages over grapheme-based and conventional phoneme-based approaches. Grapheme-based approaches are mainly based on the number of identical alphabets in the two words. Phoneme-based approaches are mainly based on the pronunciation similarity between phones. In the conventional phoneme-based approaches [14, 15], the similarity scores between phones are assigned by some predefined rules which take articulatory features of phones into consideration. Instead, CSC compares two words by their digitalized physical sounds, which raise the effectiveness by embedding more discriminative information in the digitalized sound signals

Given two Chinese words $A = \{a_1 a_2 \ldots a_N\}$ and $B = \{b_1 b_2 \ldots b_M\}$ where a_n is the n^{th} character in Chinese word *A* and b_m is the m^{th} character in Chinese word *B*. *N* is not necessarily equal to *M*. A dynamic programming based approach to comparing the similarity of smallest distortion for *A* and *B* by adjusting the warp on the time axis is employed. The recurrence formula is defined as follows in which $T(N, M)$ is the similarity of $\{a_1 a_2 \ldots a_N\}$ and $\{b_1 b_2 \ldots b_M\}$, and $sim(a_n, b_m)$ is the similarity for two Chinese characters.

$$T(N,M) = \max \begin{cases} T(N-1, M-1) + sim(a_N, b_M) \\ T(N-1, M) \\ T(N, M-1) \end{cases}$$

We constructed two similarity matrices for comparing the similarity between Chinese characters [12]. One is for the 37 phonetic symbols which are used to make of the pronunciation of a Chinese character. The other is for the 412 basic sounds which include all pronunciations of Chinese characters without considering tones.

The similarity between two Chinese characters is measured by $sim(a_n, b_m) = w \times sim_{s37}(a_n.IC, b_m.IC) + (1 - w \times s\, m_{s412}(a_n, b_m)$ where $a_n.IC$ and $b_m.IC$ represent their initial consonant (*IC*). According to our experience, final sound heavily influences speech sound comparison. Therefore, we adopt 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. $sim(a_n, b_m)$ is the weighted similarity between character a_n and b_m obtained from the similarity matrices of the 37 phonetic symbols and the 412 character pronunciations. w represents a trade-off between weighting the initial consonant and the whole character and is set to 0.4 empirically. For example, the similarity between two Chinese characters 森(sen) and 生(sheng) is measured by first converting them to the representation of their corresponding phonetic symbols, namely, ㄙㄣ(sen) and ㄕㄥ(sheng), respectively. They have initial consonants ㄙ(si) and ㄕ(shi), respectively. Then, the score is calculated by the formula, sim(森,生) = sim(ㄙㄣ,ㄕㄥ) = 0.4 sim_{s37}(ㄙ, ㄕ) + 0.6 sim_{s412}(ㄙㄣ,ㄕㄥ). According the two similarity matrices s37 and s412, sim_{s37}(ㄙ,ㄕ) =0.66 and sim_{s412}(ㄙㄣ,ㄕㄥ) = 0.69. The result is 0.68, the measured similarity between two Chinese characters 森 and 生.

The normalized similarity between two words A and B which takes into account the length of the words is defined as $score_{CSC}(A,B) = T(N,M)/(0.5(N+M))$ where N and M are the lengths of A and B, respectively. The choice of normalization operation significantly influences the similarity comparison. We set it to the average length of N and M according empirical results indicated in [12]. A high score between an *CST* and the *TL* implies the *CST* is very likely an *ST* of the *TL*.

4 Experiments

We collected a total of 50 Chinese transliterations (*TLs*) from the Web. The data were drawn from two major types of proper nouns, i.e., locations and personal names. 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.

4.1 Quality of Query Strategies

Each of the 50 *TLs* was submitted to Google search engine and the first 20 snippets were collected as the core snippets of the *TL*. For each *TL*, the top five association words were used to collect various sets of the candidate snippets according to different strategies mentioned in section 2.2. Google also suggests synonymous

transliterations with respect to some user queries. We therefore consider their recommendation as well in the experiment.

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 using the query consisting of *m* associated words;
$Q_{m\text{-}AsOri}$: collecting snippets by using a query consisting of *m* associated word plus the foreign word;
Q_{GR}: Google recommendation.

The following discusses how effective each strategy is able to collect a *better* set of candidate snippets, which shall contain as many synonymous transliterations as possible.

The second row in Table 1 shows the ratio of *TL* having at least one synonym in the collected snippets under a certain query strategy. Under the strategy Q_{Ori}, the ratio is 74%; in other words, 37 out of 50 *TLs* have at least one synonym in the retrieved snippets. $Q_{2\text{-}AsOri}$ brings the best performance, which is 92%. The result also shows that only 4% has recommendation from the search engine. Among the inputs, three of the 50 *TLs* do not has any synonym in the collected snippets, including 雅典娜 (Athena), 托拉斯 (Trust) and 赫爾利 (Hurley).

Surprisingly, the combination of the original word along with association words performs better than using the original word alone. For instance, the transliterations 馬斯哈托夫 (Maskhadov), 巴薩拉 (Basra), 賽普拉斯 (Cypress), 費雪 (Fisher), 蓋亞 (Gaea), and 鮑爾 (Powell) have no *ST*s in the collected snippets by Q_{Ori}, but they do have by $Q_{2\text{-}AsOri}$ or $Q_{1\text{-}AsOri}$. The reason is that these transliterations are more popular than their synonymous transliterations. As a result, all the returned snippets, of which the number is limited to about 1000 by the search engine, by Q_{Ori} contain only the most commonly seen transliterations, no other synonymous transliterations. A *stricter* query strategy which additionally include association words along with the original foreign word help to bring the Web pages containing synonymous transliterations to the set of the returned first 1000 pages.

Second, we test how many synonymous transliterations could be retrieved in average under different methods with respect to a given *TL*. Experimental results in the third row of Table 1 show that including *Ori* along with their association words in the query outperforms their counterpart, which does not include *Ori*. Furthermore, the parameter *m* (the number of association words in a query) is better not to be greater than 3. Requesting too many association words in a snippet would limit the number of snippets that we can retrieve.

Given the 50 *TLs*, we retrieved in total 366 STs, of which 252(69%), 246(67%), 136(37%), 145(40%), 86(23%), 54(15%), 40(11%), 22(6%), 2(0.5%) by 2-AsOri, 1-AsOri, 3-AsOri, Ori, TL, 2-As, 3-As, 1-As, and GR, respectively. 322 (88%) out of 366 can be retrieved by 2AsOri and 1AsOri together.

Finally, we inspect how *uniqueness* the retrieved result of a method is, i.e., how many words which are retrieved uniquely by the method but not by the other methods.

Table 1. Probability of a *TL* having at least one synonym in the collected snippets and the average number of retrieved synonymous transliterations

Method	2-AsOri	1-AsOri	3-AsOri	Ori	TL	2-As	3-As	1-As	GR
ST. Occurrence Probability	0.92	0.9	0.82	0.74	0.50	0.50	0.38	0.36	0.04
Average number of STs	5.04	4.92	2.72	2.90	1.72	1.08	0.80	0.44	0.10
Uniqueness	0.318	0.419	0.062	0.093	0.023	0.054	0.016	0.016	0.000

Table 1 shows among those words retrieved by only one method, about 40% and 30% are by $Q_{1\text{-}AsOri}$ and $Q_{2\text{-}AsOri}$, respectively. Except for GR, other methods can retrieve more or less some unique STs.

4.2 Performance of Synonymous Transliterations Extraction

This section presents how well the confirmation model can recognize those identified candidate synonymous transliterations (*CSTs*) as true synonymous transliterations (*STs*). The evaluation measures include precision, the average number of retrieved *STs* and the inclusion rate.

Because $Q_{2\text{-}AsOri}$ was more effective in retrieving candidate snippets in which *STs* appear, we use the set of *CSTs* extracted from the candidate snippets by $Q_{2\text{-}AsOri}$ via the dynamic alignment process. The dynamic alignment approach reduced the size of the *n*-gram terms 355,943 to the size of *CST* terms 56,408. We further utilize the CSC approach [12] to measure the similarity between a *CST* term and the *TL*. The initial consonant weight is set to $w = 0.4$ which is suggested in [12]. A high score indicates high pronunciation similarity between the *CST* and the TL and implies that they are likely synonymous.

Fig. 3 shows retrieval precision and the average number of retrieved *ST*s with respect to various similarity thresholds by the CSC. The result shows that all extracted *STs* acquire at least a 0.5 CSC similarity score. It also shows that the precision is high (over 0.89) when the score is greater than 0.9.

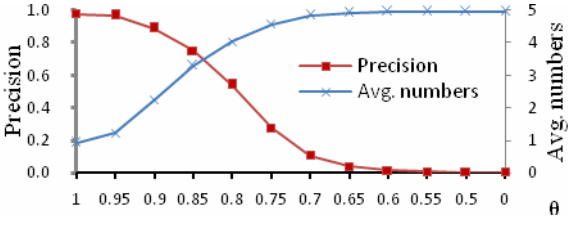

Fig. 3. Precision and average number of collected synonyms under various similarity scores by CSC

AR (average ranking), ARR (average reciprocal rank) and the inclusion rate, which are commonly used for the evaluation in information retrieval, are calculated for the data set according to the rank of the similarity score of a true *ST* to the *TL*. AR and ARR are 7.22 and 0.74, respectively. For the inclusion rate, 67% of *ST*s are included

in top-1, 81% are included in top-5, 88% are included in top-10 and 99% are included in top-100. The lowest rank of a true *ST* is 324.

5 Conclusions and Future Directions

In this paper we present a framework for collecting synonymous transliterations from the Web with respect to a given input transliteration. The research result can be applied to construct ontology of famous person names. Our method uses the online retrieved Web pages collection as the corpus. Unlike the conventional approaches in information retrieval, a manually pre-collected set of documents is used as the corpus which may engender bias. Moreover, to extract synonymous transliterations from the retrieved Web snippets, we compare the similarity between unknown words and the input transliteration by an approach based on comparing digitalized physical sounds. We will continue to improve the precision of identified synonymous transliterations in our future work.

Acknowledgement. This work is supported by National Science Council, Taiwan under grant NSC 96-2416-H-224-004-MY2.

Reference

1. Carpineto, C., Bordoni, F.U., Mori, R.D., Avignon, U.O., Romano, G., Bordoni, F.U., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems 19(1), 1–27 (2001)
2. Huang, S., Chen, Z., Yu, Y., Ma, W.-Y.: Multitype features coselection for Web document clustering. IEEE Transactions on Knowledge and Data Engineering 18(4), 448–459 (2006)
3. Cheng, P.-J., Teng, J.-W., Chen, R.-C., Wang, J.-H., Lu, W.-H., Chien, L.-F.: Translating unknown queries with Web corpora for cross-language information retrieval. In: Proceedings of ACM SIGIR, Sheffield, South Yorkshire, UK (2004)
4. Cilibrasi, R.L., Vitanyi, P.M.B.: The Google similarity distance. IEEE Transactions on Knowledge and Data Enginerring 19(3), 370–383 (2007)
5. Tsuji, K.: Automatic extraction of translational Japanese-Katakana and English word pairs from bilingual corpora. International Journal of Computer Processing of Oriental Language, 261–280 (2002)
6. Stalls, B.G., Kevin, K.: Translating names and technical terms in arabic text. In: Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages (1998)
7. Somers, H.L.: Similarity metrics for aligning children' s articulation data. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, pp. 1227–1231 (1998)
8. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: IEEE Trans. Acoustics, Speech, and Signal Proc. ASSP, pp. 43–49 (1978)
9. Lin, W.H., Chen, H.H.: Backward machine transliteration by learning phonetic similarity. In: Proceedings of the Sixth Conference on Natural Language Learning, Taipei, Taiwan, pp. 139–145 (2002)

10. Lin, W.H., Chen, H.H.: Similarity measure in backward transliteration between different character sets and its applications to CLIR. In: Proceedings of Research on Computational Linguistics Conference XIII, Taipei, Taiwan, pp. 97–113 (2000)
11. Lee, C.J., Chang, J.S., 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), 121–145 (2006)
12. 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 (2007)
13. Kuo, J.-S., Li, H., Yang, Y.-K.: A phonetic similarity model for automatic extraction of transliteration pairs. ACM Trans. Asian Language Information Processing (2007)
14. Kondrak, G.: Phonetic alignment and similarity. Computers and the Humanities 37(3), 273–291 (2003)
15. Chen, H.H., Lin, W., Yang, C.C., Lin, W.H.: Translating/transliterating named entities for multilingual information access. Journal of the American Society for Information Science and Technology, 645–659 (2006)