

Incorporating Pronunciation Variation into Different Strategies of Term Transliteration

Jin-Shea KUO^{1,2}

Ying-Kuei YANG²

¹Chung-Hwa Telecommunication Laboratories, 12, Lane 551, Sec. 5, Min-Tsu Rd., Yang-Mei, 326, Taoyuan, Taiwan
jskuo@cht.com.tw

²Electrical Engineering Dept., National Taiwan University of Science and Technology, 43, Sec. 4, Keelung Rd., 106, Taipei, Taiwan
ykyang@mouse.ee.ntust.edu.tw

Abstract

Term transliteration addresses the problem of converting terms in one language into their phonetic equivalents in the other language via spoken form. It is especially concerned with proper nouns, such as personal names, place names and organization names. Pronunciation variation refers to pronunciation ambiguity frequently encountered in spoken language, which has a serious impact on term transliteration. More than one transliteration variants can be generated by an out-of-vocabulary term due to different kinds of pronunciation variations. It is important to take this issue into account when dealing with term transliteration. Several models, which take pronunciation variation into consideration, are proposed for term transliteration in this paper. They describe transliteration from various viewpoints and utilize the relationships trained from extracted transliterated-term pairs. An experiment in applying the proposed models to term transliteration was conducted and evaluated. The experimental results show promise. These proposed models are not only applicable to term transliteration, but also are helpful in indexing and retrieving spoken document retrieval.

1. Introduction

Machine transliteration plays an important role in machine translation. The importance of term transliteration can be realized from an analysis of the terms used in 200 qualifying sentences that were randomly selected from English-Chinese mixed news pages. Each qualifying sentence contained at least one English word. Analysis showed that 17.43% of the English terms were transliterated, and that most of them were content words (words that carry essential meaning, as opposed to grammatical function words such as conjunctions, prepositions, and auxiliary verbs) (Kuo, 2004). Lee (1998) also reported a similar result about the ratio of transliterated-term in Korean KTSET 2.0 corpus.

Generally, a transliteration process starts by first examining a pre-compiled lexicon composing of many transliterated-term pairs collected manually or automatically. If a term can be found in the lexicon, one of the transliterated equivalents is selected; otherwise, the transliteration system then deals with this out-of-vocabulary term to try to generate a transliterated term via a sequence of pipelined conversions (Knight, 1998). A phoneme-based conversion approach was used in this pipelined procedure to handle English-Japanese transliteration. Kang (2000) proposed an approach using text-to-text direct correspondence for Korean-English transliteration. By utilizing the combined information of text-to-text and phoneme-based correspondences in converting basic pronunciation units or components of pronunciation units, Lee (1998) and Bilac (2004) proposed approaches to handle English-Korean and Japanese-English transliteration problems, respectively.

Pronunciation variation has often been encountered in daily conversations and, therefore, in transliteration. This issue has been discussed in speech recognition (Riley, 1991), but has not been discussed extensively with respect to term transliteration. Two pronunciation variations are encountered when transliterating a source language term into its target language counterpart. One occurs at the source language side, and the other one happens during the transliteration process. For example, elision is quite common in English speech. /t/ is often elided before consonants or when they are parts of a sequence of two or three consonants (Jurafsky, 2000). Another example is an isolated pronunciation unit, such as /l/ of /poldə/, which is converted from “polder” using a letter-to-phoneme system, may or may not be transliterated into a Chinese equivalent depending on whether you pronounce this liquid syllable swiftly or not. This apparently random phenomenon can be referred to transliteration pronunciation variation. Two transliterated terms, “波德 (bo-de)” and “波爾德

(bo-er-de),” can be obtained when dealing with “polder” transliteration. The final one depends on whether /l/ is frequently mapped to “爾(er)” or null; however, both terms are correct for term transliteration. Taking pronunciation variations into account when modeling transliteration is the main focus of this paper. Before this issue can be dealt with, a transliteration model must be trained with exposure to a large quantity of transliterated-term pairs. More than 200,000 transliterated-term pairs were extracted using different approaches (Kuo, 2003; Kuo, 2004). These paired terms, which reflected real cases, can be used as a training corpus in studying term transliteration.

English and Chinese are two of the most commonly used languages around the world and have many different features. For example, each word in Chinese is monosyllabic; on the other hand, many English words have multiple syllables. To convert phonemes statistically between two languages that belong to different language families is not an easy task when dealing with term transliteration. It is even more difficult if the pronunciation variation issue is taken into account.

A modular learning algorithm has been proposed for English-Japanese term transliteration (Knight, 1998). This algorithm requires all the specifications of relationships between source language texts and their phonetic representation, source language phonemes and the target language counterparts, and the target language phonemes and their textual data. Thus a source language letter-to-sound system, a cross-language phoneme conversion, and a target language sound-to-letter mapping are required to transliterate a term cross-linguistically. If any sub-system is not available, it is impossible to transliterate a term using this algorithm. However, this algorithm establishes a conceptual foundation for term transliteration. Statistical models for machine translation have been proposed in Brown (1993), and a noisy channel model for spelling correction has been proposed in Brill (2000). These models can be adapted to take pronunciation variation, which has not been discussed extensively when dealing with term transliteration, into consideration.

Transliteration models for cases where one or more required components in the pipelined procedure are not available, at the same time taking pronunciation variation into account, are proposed in this paper. These models are still able to transliterate a term if a source language letter-to-sound system and/or a target language sound-to-letter mapping may be missing.

The remainder of this paper is organized as follows. Section 2 describes the proposed models for term transliteration. Experimental results are presented in Section 3. Section 4 discusses on term transliteration. Conclusions are drawn in Section 5.

2. The Proposed Models

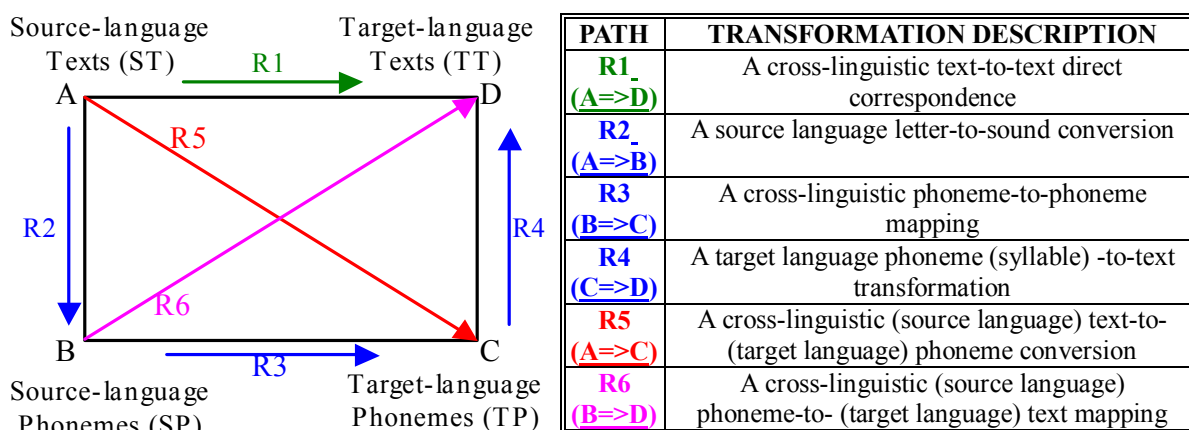


Figure 1. Different models for transliteration (paths: R1, R2=>R3=>R4, R2=>R6 and R5=>R4).

Several models, taking pronunciation variation into account, for term transliteration are proposed and depicted in Figure 1. The source language and the target language referred to in this paper are English and Chinese, respectively. In Figure 1, each side in the central rectangle represents a transformation.

The modular learning procedure mentioned above requires R2, R3 and R4 transformations to complete a term transliteration. Several variants, such as R5=>R4 path (referred to as LTPL), R2=>R6

path (referred to as LSPL), and R1 path (referred to as LL), can be obtained if any transformation in this procedure is not available and can be observed from Figure 1. In addition to these variants, the R2=>R3=>R4 path (referred to as LPPL) shown in Figure 1, is the most similar to the original pipelined procedure. LPPL is a model for term transliteration with three components, namely, a source language letter-to-sound system, a cross-linguistic phoneme-to-phoneme mapping and a target language phoneme-to-text transformation. One or more of these three components could be missed. If a source language letter-to-sound system is not available, then LPPL is reduced to a LTPL model. If a target language phoneme-to-text conversion missed, then a LSPL algorithm is used. If all three components missed, then a LL procedure with a cross-linguistic text-to-text correspondence is used.

These procedures for term transliteration first syllabify texts or phonemes and then map these text or phoneme syllables to suitable representations using the relationships described in Fig. 1. Two syllabification algorithms, text-based syllabification algorithm (TSA) and phoneme-based syllabification algorithm (PSA), are used in term transliteration. LPPL and LSPL use PSA; on the other hand, LTPL and LL use TSA. The PSA, which is similar to the algorithm described in Jurafsky (2000) with minor modifications, is the same as that described in Kuo (2004) and the TSA used is described briefly. An English word is often composed of multiple syllables; whereas, a Chinese word is monosyllabic. First, each English character in an English term is identified as a vowel, a nasal consonant or a consonant. For example, the characters “a”, “n” and “b” are viewed as a vowel, a nasal consonant and a consonant, respectively. Second, consecutive characters of the same attribute form a cluster. However, some characters, such as “ch”, “ng” and “ph”, always combine together to form complex consonants. Such complex consonants are also taken into account in the syllabification process. A Chinese syllable is composed of an initial and a final. An initial is similar to a (nasal) consonant in English, and a final is analogous to a vowel or a combination of a vowel and a nasal consonant.

A transliteration equivalent \check{T} in the target language with the largest probability can be selected for each token \check{S} in the source language using the Bayes rule. Suppose that the possible transliterated-token pair is denoted by

$$\hat{J}=(\check{S}, \check{T}) \equiv \arg \max _{W_t} p\left(W_t \mid W_s^i\right), \quad (1)$$

where W_t and W_s^i are tokens of the target language and the source language, respectively, and a transliteration of a sequence of tokens can be cascaded by token transliterations

The details of describing how to determine \hat{J} by equations of each proposed transliteration algorithms, LPPL, LTPL, LSPL and LL, are described as follows.

1. LPPL (R2=>R3=>R4): Basically, LPPL is an algorithm converting terms in source language into their phonetic equivalents in the target language via spoken form. This is the approach most similar to the original modular learning algorithm conceptually. If all the components required for term transliteration are available, \hat{J} can be determined by means of equation (2):

$$\hat{J} \approx \arg \max _{W_t} p\left(H_s^i \mid W_s^i\right) p\left(H_t \mid H_s^i\right) p\left(W_t \mid H_t\right), \quad (2)$$

where H_s^i and H_t are phonemes converted from terms in source language and target language, respectively. If an existing source language letter-to-sound system is used and the target language phoneme-to-text conversion is deterministic, the cross-linguistic phoneme-to-phoneme conversion is the main focus in this algorithm. Taking pronunciation variation into consideration, the source language phonemes, $H_s^i = \{(H_s^{i11}, \dots, H_s^{in_1}), \dots, (H_s^{iU1}, \dots, H_s^{iUn_U})\}$, may contain of many different combinations in which syllables with isolated and elided consonants may or may not be silent in transliteration. There are k syllables and U sub-sets of syllables in H_s^i in total and the items in each sub-set are sorted in descending order in indexes. Each sub-set of the source-language syllables is a basic unit used to transliterate into a term in the target language. The window size of the target-language syllables varies according to the size of the selected basic unit. The syllable-to-syllable

probability then can be estimated by a set of context-dependent syllables trained from the transliteration lexicon constructed by Kuo (2004). The main focus of equation (2) can be expressed by means of equation (3):

$$\begin{aligned} \arg \max_{H_t} p(H_t | H_s^i) &\approx \arg \max_{H_t^j H_s^{iu}} p(H_t^1 H_t^2, \dots, H_t^n | H_s^{i1} H_s^{i2}, \dots, H_s^{ik}) \\ &\approx \arg \max_{H_t^j H_s^{iu}} \prod_{w=1}^{M_{iu}} (p(H_t^{jw} | H_s^{i(m-1)}, H_s^{im}, H_s^{i(m+1)}) + \varepsilon), \end{aligned} \quad (3)$$

where $M_{iu} = |H_s^{iu}|$ is the window size of each sub-set and m is the absolute syllable index of w in H_s^i and ε is a very small constant used to avoid any zero probability. A null syllable is attached for the cases of those syllables at the beginning and end of the pronunciation units. According to our definition of syllabifying English syllables, there are thousands of different English syllables in total. In order to alleviate the data sparseness problem, the probability estimation of using trigrams in equation (3) and can be reduced to that using bi-grams.

The phoneme-based confusion matrix provides a fine-grained control that is used when the quality of the syllable-based confusion matrix is not good enough. A Chinese syllable can be divided into initial and final parts. A Chinese initial is almost the same as a consonant cluster in English, and a Chinese final is also analogous to an English vowel or a combination of a vowel and a final consonant cluster in terms of functionality. Suppose that the initial part and the final part are generated independently. Equation (3) can be estimated using these low-level primitives and can also be estimated by equation (4):

$$\begin{aligned} \arg \max_{H_t} p(H_t | H_s^i) &\approx \arg \max_{H_t^j H_s^{iu}} \prod_{w=1}^{M_{iu}} (p(I_t^{jw} | I_s^{iuw}) p(F_t^{jw} | F_s^{iuw}) + \varepsilon) \\ &\approx \arg \max_{H_t^j H_s^{iu}} \prod_{w=1}^{M_{iu}} (p(HI_t^{jw} | HI_s^{i(m-1)}, HI_s^{im}, HI_s^{i(m+1)}) p(HF_t^{jw} | HF_s^{i(m-1)}, HF_s^{im}, HF_s^{i(m+1)}) + \varepsilon), \end{aligned} \quad (4)$$

where I_t^{jw} and F_t^{jw} are sets of initial (HI_t^{jw}) and final (HF_t^{jw}) parts of H_t^{jw} , respectively, and I_s^{iuw} and F_s^{iuw} are sets of the initial (HI_s^{iuw}) and the final (HF_s^{iuw}) parts of H_s^{iuw} , respectively.

Equations (3) and (4) can be combined to form an equation with different weights for syllable-based and phoneme-based confusion matrices. By changing the weights, we can control the effects of different confusion matrices on term transliteration.

2. LTPL (R5=>R4): If a source language letter-to-sound system is not available, then cross-linguistic rules for converting source language terms into target language phonemes can be obtained from the extracted transliterated-term pairs. Transliterating a source-language term into a target-language equivalent can be achieved using a cross-linguistic (source language) text-to- (target language) phoneme correspondence (R5) and a target language phoneme-to-text conversion (R4) together. This transliteration procedure can be expressed using equation (5):

$$\hat{J} \approx \arg \max_{W_t} p(H_t | W_s^i) p(W_t | H_t). \quad (5)$$

Suppose that the relationship between target language phoneme-to-text conversion is one-to-one, the cross-linguistic text-to-phoneme conversion is the main focus in this algorithm. If pronunciation variation is taken into account, the cross-linguistic text-to-phoneme of the term transliteration procedure can be expressed using equation (6).

$$\begin{aligned} \arg \max_{H_t} p(H_t | W_s^i) &\approx \arg \max_{H_t^j W_s^{iu}} p(H_t^1 H_t^2, \dots, H_t^n | W_s^{i1} W_s^{i2}, \dots, W_s^{ik}) \\ &= \arg \max_{H_t^j W_s^{iu}} \prod_{w=1}^{M_{iu}} (p(H_t^{jw} | W_s^{i(m-1)}, W_s^{im}, W_s^{i(m+1)}) + \varepsilon), \end{aligned} \quad (6)$$

where $M_{iu} = |W_s^{iu}|$ is the window size of each sub-set in W_s^i . Equation (6) determines the goodness of transliterated-token pairs based on the degree of cross-linguistic text-to-phoneme similarity.

The advantage of this algorithm is that it gets rid of the source-language letter-to-sound system in the transliteration process. It is possible that a researcher cannot obtain a sufficiently large quantity of letter-phoneme pairs like the CMU pronunciation dictionary¹ in an unfamiliar language. This algorithm can work around the lack of a source-language pronunciation dictionary.

3. LSPL (R2=>R6): If a target language phoneme-to-text conversion (R4) is not available, then cross-linguistic rules (R6) for converting source language phonemes into target language texts can be estimated from the training corpus. Combining a source language letter-to-sound system (R2) and R6, yields a new transliteration procedure. This procedure can be expressed using equation (7).

$$\hat{J} \approx \arg \max_{W_t} p(H_s^i | W_s^i) p(W_t | H_s^i) \quad (7)$$

The advantage of this algorithm is that researchers unacquainted with the target language can still transliterate a term from source language to target language without relying on the conversion between phoneme-to-text in the target language. Suppose that an existing source language letter-to-sound system is used and the cross-linguistic phoneme-to-text mapping is the main focus. If pronunciation variation is taken into account, the cross-linguistic phoneme-to-text of the term transliteration can be expressed using equation (8):

$$\begin{aligned} \arg \max_{W_t} p(W_t | H_s^i) &\approx \arg \max_{W_t^j H_s^{iu}} p(W_t^1 W_t^2, \dots, W_t^n | H_s^{i1} H_s^{i2}, \dots, H_s^{ik}) \\ &= \arg \max_{W_t^j H_s^{iu}} \prod_{w=1}^{M_{iu}} (p(W_t^{jw} | H_s^{i(m-1)}, H_s^{im}, H_s^{i(m+1)}) + \varepsilon), \end{aligned} \quad (8)$$

where $M_{iu} = |H_s^{iu}|$ is the window size of each sub-set in W_s^i . Equation (8) determines the goodness of transliterated-token pairs based on the cross-linguistic phoneme-to-text similarity degree.

4. LL (R1): if all of R2 (a source language letter-to-sound system), R3 (a cross-linguistic phoneme-to-phoneme mapping) and R4 (a target language phoneme-to-text transformation) are not available, then transliterating a term is also possible. In such case, a cross-linguistic text-to-text correspondence using the rules mined from the training corpus can be used. One of the problems of this algorithm is that a large corpus of transliterated-term pairs is required because the text-based syllabification algorithm used here for mapping texts is simple, intuitive and coarser than the phoneme-based syllabification algorithm. The text-to-text mapping of this transliteration can be estimated directly using equation (1). This algorithm is similar to (Kang; 2000), however, the basic pronunciation unit used here is based on syllables and the correspondence is learned from a large quantity of paired terms, which reflect real cases of term transliteration. If pronunciation variation is taken into account, the direct text-to-text correspondence can be re-expressed using equation (9):

$$\begin{aligned} \arg \max_{W_t} p(W_t | W_s^i) &\approx \arg \max_{W_t^j W_s^{iu}} p(W_t^1 W_t^2, \dots, W_t^n | W_s^{i1} W_s^{i2}, \dots, W_s^{ik}) \\ &= \arg \max_{W_t^j W_s^{iu}} \prod_{w=1}^{M_{iu}} (p(W_t^{jw} | W_s^{i(m-1)}, W_s^{im}, W_s^{i(m+1)}) + \varepsilon), \end{aligned} \quad (9)$$

where $M_{iu} = |W_s^{iu}|$ is the window size of each sub-set in W_s^i . Equation (9) determines the goodness of transliterated-token pairs based on the degree of cross-linguistic text-to-text similarity.

This algorithm makes sense when a large quantity of paired transliterated-terms is available. Kuo (2004) extracted many transliterated-term pairs and has laid a solid foundation for such algorithm. The

¹ <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

experimental results shown later reveal that this algorithm is viable.

These proposed models taking pronunciation variation into account described term transliteration from various viewpoints and could be used to transliterate an out-of-vocabulary term when one or more required components are not available. Even if only the paired transliterated-terms are available, a cross-linguistic text-to-text mapping can be used to transliterate a new term. This is a very important feature because not all the sub-systems required to perform a term transliteration can be obtained when dealing with a pair of languages which researchers are not familiar with.

M_{lppl} , M_{lspl} , M_{ltpl} and M_{ll} are models used to estimating probabilities for determining transliterated terms using LPPL, LSPL, LTPL and LL algorithms, respectively. Combining all models described above together, equation (1) can be re-expressed using equation (10), and changing the weights of models different transliteration behaviors can be deduced.

$$\hat{J} \approx \arg \max_{w_i} (\rho M_{lppl} + \delta M_{lspl} + \varphi M_{ltpl} + \phi M_{ll}), \rho + \delta + \varphi + \phi = 1. \quad (10)$$

3. Experimental Results

A training corpus composed of about 45,000 transliterated-term pairs was used to generate the necessary relationships for term transliteration. A total of 279 personal names were fetched from a visitor list². Personal names in many different languages appeared in this list. Some examples of original transliterations in the list are shown in Table 1. The transliterated source-language words are underlined. It can be seen that not all parts of a name were transliterated and this will complicate the transliteration judgments. More efforts on exploring rules for transliterating personal names are required. The Chinese transliterated-terms extracted from the visitor list were viewed as the reference patterns when judging manually the equivalents transliterated by the proposed methods.

Table 1. Examples of term transliterations selected from the testing corpus.

English Terms	Salote Mafile'o <u>Pilolevu</u> Tuita	Stefano <u>de Luca</u>	Ismat D. <u>El-kurdi</u>	Brown <u>Mpinganjira</u>
Chinese Transliterated Terms	碧露麗富	戴魯卡	科迪	曠甘紀

Fifty personal names, each of them has one or more isolated syllables to be or not to be rendered, in the list were randomly selected and transliterated. Transliteration using the previously described models, LPPL (R2=>R3=>R4), LSPL (R2=>R6), LTPL (R5=>R4) and LL (R1) in direct transliteration style, with or without taking pronunciation variation into consideration were conducted. The direct transliteration style used here is to transliterate every source-language syllable into a word or a syllable in target language. All the terms are viewed as out-of-vocabulary ones and there is no table look-up during the term transliteration process. Some English terms transliterated by proposed models with or without taking pronunciation variation (PV) into account are depicted in Table 2.

Table 2. Chinese transliterated equivalents of “Richard Prebble” taking or not taking pronunciation variation into account.

	With PV	Without PV
LL	理查德·普萊布爾	理查德·普萊布爾
LPPL	利查德·普來勃	利查德·普來勃爾
LSPL	理查德·普萊博	理查德·普萊博爾
LTPL	利查德·普萊布爾	利查德·普萊布爾

Three testers, P, H, and T verified the produced results manually. The verified transliterated terms were divided into four classes: wrong (type I), almost wrong (type II), almost correct (type III) and correct (type IV). Table 3 records the number of experimental transliterations scored as Type III and

²http://www.mofa.gov.tw/almanac98/section_3/page3-5.htm

Type IV, with the two categories separated by a '/'. After assigning weights of 0.0, 0.0, 0.75 or 1 to a transliterated term of type I, II, III, or IV, the average scores of the results verified by each tester are also displayed in parentheses. The average scores of the transliteration results produced by all models with and without taking pronunciation variation into account are 0.9317 and 0.9267, respectively. Examining the performances achieved by text-based syllabification algorithm and phoneme-based syllabification algorithm, the average scores of the results produced by using text-based syllabification algorithm (LL and LTPL) and phoneme-based syllabification algorithm (LPPL and LSPL) are 0.9275 and 0.9258 and 0.9292 and 0.9342 in Table 3, respectively.

Table 3. The transliteration results produced using models with or without taking pronunciation variation into account.

	P	H	L	P (PV)	H (PV)	L (PV)
LL	19/31(0.905)	16/34(0.920)	11/39(0.945)	17/33(0.915)	16/34(0.920)	15/35(0.925)
LPPL	9/41(0.955)	17/33(0.915)	18/32(0.910)	7/43(0.965)	16/34(0.920)	16/34(0.920)
LSPL	9/41(0.955)	20/30(0.900)	16/34(0.920)	6/44(0.970)	18/32(0.910)	16/34(0.920)
LTPL	16/34(0.920)	15/35(0.925)	10/40(0.950)	12/38(0.940)	14/36(0.930)	11/39(0.945)

4. Discussion

From experimental results showed that those algorithms taking pronunciation variation into account achieved better performances than that generated by algorithms without considering pronunciation variation. Several points can be observed from the experiments:

1. A target-language term should have pronunciation as close to the source-language transliterated equivalent as possible.
2. Term transliteration addresses the problem of converting terms in one language into their phonetic equivalents in the target language via spoken form. The phonetic equivalence has been the focus in studying term transliteration. From the experimental results showed that the characters used also play an important role in term transliteration, especially when dealing with personal name transliteration. Good transliterated equivalents not only have phones very close to the original terms but also have suitable characters to represent the original ones. This is because each character has unique meanings, which may imply good, peaceful or bad, in Chinese.
3. Transliterated equivalents, which have been frequently used in daily conversation, should be used in term transliteration. For example, "James Bond" has been transliterated into "詹姆士(zhan-mu-shi) · 龐德(pang-de)" in Chinese. Though both pronunciations are very close and even that the same, if the same term is transliterated into "展木世(zhan-mu-shi) · 磅得(pang-de)", it may not be accepted by people who have been familiar with the Chinese transliterated counterpart.
4. Terms may also be transliterated into Chinese in different ways. "Vincent Sarubbi" was transliterated into "文森特 · 沙魯比" (wen-shen-te · sha-lu-bi) in a direct way; however, the transliterated counterpart of this term extracted from the Web page is "沙文森" (sha-wen-shen) transliterated according to the Chinese naming conventions. A Chinese name can be decomposed into two parts, a family name and a given name, which is similar to the last name and the first name according to the western naming conventions, respectively. According to the Chinese naming conventions, the transliterated term, "沙文森", can be decomposed into two parts, namely, a Chinese family name, "沙", which was obtained by rendering the first syllable of the lat name part, and a given name, "文森", which was transliterated by taking pronunciation variation into account with a elided /t/. Comparing these two kinds of transliteration ways, the terms transliterated from foreign terms into Chinese in a direct way are lengthy and tedious; on the other hand, the terms transliterated in Chinese style are short and implies some meanings in them. It will be an interesting issue to transliterate foreign names into Chinese in a Chinese way and will be addressed in the future.

5. Another noteworthy point is those algorithms using text-based syllabification algorithm achieved performances very close to that produced using phoneme-based syllabification algorithm. This phenomenon can be observed from Table 3. It means that if many transliterated-term pairs are obtained then a letter-to-sound system for the source-language terms is not necessarily required. This will make transliteration easier for other language pairs.
6. All the relationships described in Figure 1 can also be used in spoken document retrieval (Meng, 2001; Virga, 2003) and text-to-speech synthesis. These relationships can provide information for cross-linguistic indexing. It is also helpful to transliterate and synthesize unknown proper nouns in mixed-language text-to-speech synthesis.
7. To judge whether a transliterated term with some syllables may or may not be elided is good or not is not an easy task. Another evaluation metric incorporating the importance of characters used to represent transliterated equivalents may be addressed and used.

5. Conclusions

Pronunciation variation is a phenomenon of pronunciation ambiguity frequently encountered in speech tasks, which has an important impact on term transliteration. Several models for term transliteration, which fully utilize various relationships explored from transliterated-term pairs and also take pronunciation variation into account, are proposed in this paper. An experiment on English-to-Chinese term transliteration was conducted and evaluated using the proposed models. From the experimental results showed that the proposed algorithms taking pronunciation variation into account achieved better performances than that without considering pronunciation variation. Observations from the results also showed that term transliteration also has to take characters, which are used to represent the transliterated-term, into consideration in addition to the naïve conversion of terms in one language into their phonetic equivalent via spoken form. The proposed models are not only applicable to term transliteration, but also are helpful to spoken document retrieval.

References

- Brill, E. and R. C. Moore 2000, An Improved Error Model for Noisy Channel Spelling Correction, In *Proceedings of 38th ACL*.
- Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer 1993, The Mathematics of Statistical Machine Translation: Parameter Estimation, *Computational Linguistics*, Vol. 19, No. 2, pp. 263-311.
- Kang, In-Ho and GilChang Kim 2000 “English-to-Korean Transliteration Using Multiple Unbounded Overlapping Phoneme Chunks”, In *Proceedings of COLING*, pp. 418-424.
- Jurafsky, D. and J. H. Martin 2000. *Speech and Language Processing*, pp. 102-120, Prentice-Hall, New Jersey.
- Knight, K. and J. Graehl 1998. Machine Transliteration, *Computational Linguistics*, Vol. 24, No. 4, pp. 599-612.
- Kuo, J. S. and Y. K. Yang 2003. Automatic Transliterated-term Extraction Using Confusion Matrices from Non-parallel Corpora, In *Proceedings of ROCLING XV Computational Linguistics Conference*, pp.17-32.
- Kuo, J. S. and Y. K. Yang 2004 Constructing Transliteration Lexicons from Web Corpora, In *Proceedings of 42nd ACL*.
- Lee, Jae-Sung and Key-Sun Choi 1998, English to Korea Statistical Transliteration for Information Retrieval, *Computer Processing of Oriental Languages*, Vol. 12, No. 1, pp. 17-37.
- Meng, H., W. K. Lo, B. Chen and K. Tang 2001. Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval," In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Trento, Italy.
- Pagel, V., K. Lenzo, and A. Black 1998. Letter to Sound Rules for Accented Lexicon Compression, In *Proceedings of ICSLP*, pp. 2015-2020.
- Riley, Michael D. 1991, A Statistical Model for Generating Pronunciation Networks, In *Proceedings of ICASSP (International Conference on Acoustic, Speech, and Signal Processing)*, Vol. 2, pp. 737-740.
- Bilac, S. and Hozumi Tanaka 2004, Improving Back-transliteration by Combining Information Sources, In *Proceedings of IJCNLP*, pp. 542-547, 2004.
- Virga, P. and Khudanpur S. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval, In *Proceedings of ACL Workshop on Multilingual and Mixed Language Named Entity Recognition*, pp57-64.