Recognizing Transliteration Equivalence for Enriching Domain-Specific Thesauri

Jong-Hoon Oh
Computational Linguistics Group,
Information and Network Systems Department,
National Institute of Information and Communications Technology,
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
rovellia@nict.go.jp

Key-Sun Choi
Computer Science Division, Dept. of EECS,
Korea Advanced Institute of Science and Technology (KAIST)/KORTERM/BOLA,
373-1 Guseong-Dong, Yuseong-Gu, Daejeon, 305-701, Republic of Korea
kschoi@cs.kaist.ac.kr

Abstract

Transliteration is used to translate proper names and technical terms especially from languages in Roman alphabets to languages in non-Roman alphabets such as from English to Korean, Japanese, and Chinese. "Transliteration equivalence" refers to a set of the same words that include all possible transliterated forms and the original word. Many Korean domain-specific terms are composed of transliterations. Therefore, handling transliterations and their transliteration equivalence is essential to constructing and enriching Korean domain-specific thesauri. In this paper, we propose an algorithm recognizing transliteration equivalence or transliteration pairs in domain-specific dictionaries using machine transliteration. Machine transliteration can serve as one of components in a transliteration pair acquisition method by offering a machine-generated transliterated form. Because, transliteration pair acquisition task is to find phonetic cognate in two languages, it is important to phonetically convert words in one language to that in the other language, like machine transliteration, to compare the phonetic equivalence. Our method shows about 99% precision and 73% recall rate.

Introduction

With the advent of new technology and the flood of information in WWW, it has become quite common to adopt a foreign word into one's language. The adoption is usually a process of adjusting its original pronunciation to suit the phonological regularities in the target language, along with modification of its orthographical form. This procedure of phonetically "translating" foreign words is called *transliteration*. For example, English word *data* is transliterated into Korean as 'de-i-teo'[1]. Transliteration is used to translate proper names and technical terms especially from languages in Roman alphabets to languages in non-Roman alphabets such as from English to Korean, Japanese, and Chinese. "Transliteration equivalence" refers to a set of the same words that include all possible transliterated forms and the original word. For example, a set composed of English word *data* and its Korean transliterations 'de-i-ta' and 'de-i-teo' is the transliteration equivalence. Here, 'de-i-teo' is the standard transliteration for English word *data* and 'de-i-ta' is a transliteration variation. Transliteration variations are defined as transliterations in the transliteration equivalence, which are not the

[1] In this paper, Korean transliterations are written in their Romanization form with a quotation mark (''). '-' represents a syllable boundary.

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 231–237.

standard transliteration. However, it is difficult to distinguish the standard transliteration and transliteration variations for transliterations of coined terms. Therefore the two are not distinguished in this paper.

Many Korean domain-specific terms are composed of transliterations (Oh & Choi, 2003). For example, Korean biological terms, ‘a-mil-la-a-se’ and ‘a-de-nil peb-ti-da-a-se’ are transliterations of *amylase* and *adenyl peptidase*, respectively. Therefore, handling transliterations and their transliteration equivalence is essential to constructing and enriching domain-specific Korean thesauri. In this paper, we propose an algorithm recognizing transliteration equivalence or transliteration pairs in domain-specific dictionaries using machine transliteration. The goal of our method is to find transliteration equivalence from English-Korean translation pairs, which are entries of domain-specific dictionaries. Machine transliteration can serve as one of components in a transliteration pair acquisition method by offering a machine-generated transliterated form. Because, the transliteration pair acquisition task is to find phonetic cognate in two languages, it is important to phonetically convert words in one language to that in the other language, like machine transliteration, to compare the phonetic equivalence.

This paper organized as follows. In section 1, we will describe the previous works. Section 2 shows details of our method. Section 3 deals with experiments.

1 Previous works

Recently, many researchers have been interested in an automatic method for transliteration pair acquisition, especially English-to-Japanese (Brill et al., 2001; Collier et al., 1997; Tsujii, 2002).

Collier et al.(1997) aimed at extracting proper names. He proposed a two-step procedure for the task. The first step extracts candidates for transliteration pairs. English words whose first letters are in upper cases and Japanese words written in ‘katakana’ are extracted. The second one makes a link between English and Japanese candidates through NPT transcription (Japanese-to-English phonetic conversion). Japanese candidates are transformed into NPT transcription then English-Japanese TPs are found by comparing similarity between English words and NPT transcription. This method shows 82% precision and 75% recall.

Tsuji (2002) proposed an English-Japanese transliterated pair acquisition method by extending the Collier’s method. He did not restrict target words to proper names, and he devised a string match measure based on Dices coefficient. Moreover, he trained the transliteration rules observed in the training corpora. The method achieved 83~100% precision at 75% recall.

Brill et al., (2001) proposed a statistical model for the English-Japanese transliteration pair acquisition task. He adopted the noisy-channel error model. The method employed a trainable edit distance function to find <katakana, English> pairs that have a high probability of being equivalent.

There are two different points between our method and the previous ones. The first one is caused by difference between Korean and Japanese. In Japanese, there is a character set for representing loan words or transliterations, called ‘katakana’, though words written in ‘katakana’ are not always transliterated words. Therefore, it is an easy task to recognize transliterations in Japanese by just finding words in ‘katakana’. On the contrary, Korean transliterations can not be easily recognized by just looking through texts, because pure Korean words and transliterated words share the same character set. This makes it difficult to recognize transliterations in Korean. Therefore, an algorithm is necessary to detect and recognize Korean transliterations in Korean texts.

Second one is caused by the method for converting words in one language to phonetically equivalent words or string in the other language. In other words, the phonetic conversion procedure from Japanese ‘katakana’ to English spelling, which the previous works used, is called back-transliteration in the context of Knight & Graehl (1997), while ours is transliteration (English-to-Korean). An approach with the transliteration method has an advantage over that with back-transliteration method. Because of the information losing aspect of transliteration, the invert procedure, (i.e. back-transliteration) is harder than transliteration (Knight & Graehl, 1997). This means that transliteration more correctly generates phonetically equivalent string than back-transliteration.

2 Method

Our proposed method extracts English-Korean transliteration pairs (EKTPs) through four steps. In the first step called "Extracting EKTP candidates step", the system filters out E-K translation pairs, in which Korean parts do not contain transliterations among all E-K translation pairs in domain specific dictionaries. The second step called "Machine transliteration step" transliterates English words into Korean. In the third step called "Comparing phonetic similarity step", comparing phonetic similarity between English word and Korean word makes it possible to recognize relevant EKTPs among EKTP candidates. In this step, we convert Korean words and Korean transliterated forms of English words into phonetic code represented with Korean characters.

2.1 The First Step: Extracting EKTP Candidates

In this step, we use HMM based model for detecting/recognizing transliteration model proposed by (Oh & Choi, 2003). The main idea of recognizing a Korean transliteration is that the composition of transliterations would be different from that of pure Korean words, because the phonetic system for the Korean language is different from that for the foreign language. Especially, several English consonants that occur frequently in English words, such as 'p', 't', 'c', and 'f', are transliterated into Korean consonants 'p', 't', 'k', and 'p', respectively. These consonants do not occur frequently in pure Korean words. This can be an important clue for extracting transliterated foreign words from Korean texts. For example, in a word phrase 'si-seu-tem' (system), the syllable 'tem' has high probability to be a syllable in transliterations because the consonant, 't', in the syllable 'tem' is usually not used in a pure Korean word.

For a given word phrase, Oh & Choi's algorithm tags each syllable in the word phrase with either 'F' or 'K' (a syllable with tag 'F' means that the syllable is part of a transliteration, and a syllable with tag 'K' means that the syllable is part of a pure Korean word). For example, word phrases 'si-seu-tem+eun (system + topical marker)' and 'a-de-nil peb-ti-da-a-se' (adenyl peptidase) can be tagged as "si/F + seu/F + tem/F + eun/K" and "a/F + de/F + nil/F peb/F + ti/F + da/F + a/F + se/F", by their algorithm. A series of 'F' tags makes it possible to detect and extract transliterated foreign words in the tagged results. If there is a series of 'F' tags in the result, we can determine that a given word phrase contains transliterated words and the words corresponding to the series of 'F' tags can be extracted as transliterated words.

In this step we regard E-K translation pairs as EKTP candidates, when the whole Korean words in E-K translation pairs are transliterations like 'a-de-nil peb-ti-da-a-se'. Note that the E-K translation pair <*deamino*, 'tal(脫)-a-mi-no'> will be discarded, while <adenyl peptidase, 'a-de-nil peb-ti-da-a-se'> will be selected as an EKTP candidate.

2.2 The Second Step: English-Korean Machine Transliteration

In this step, we use grapheme- and phoneme based transliteration model for English-to-Korean machine transliteration proposed by Oh & Choi (2005). The transliteration model transforms English words into Korean transliterations with machine learning algorithms. In this step all English words in EKTP candidates are transliterated into Korean.

2.3 The Third Step: Comparing Phonetic Similarity

Let *K* and *E* be Korean words and English words in EKTP candidates, respectively, and *TK* be a transliteration of *E* produced by the machine transliteration step. In this step, relevant EKTPs are selected by comparing phonetic similarity between K and TK. For the comparison, consonants in *K* and *TK* are converted into phonetic code (Consonant-to-Phonetic Code conversion). During the conversion, consonants are substitute into phonetic codes, called KODEX code[2] using the mapping table as described in table 1 (Kang, 2001), while vowels are not converted. The conversion method is similar to SOUNDEX algorithm (Odel & Russell, 1918). Let the converted phonetic code of *T* and *TK* be *T'* and *TK'*. We devise a phonetic similarity measure based on Levenshtein Distance. Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (*s*) and the target string (*t*). The distance is the number of deletions, insertions, or substitutions required to transform *s* into *t* (Levenshtein, 1965). Let *LD(T',TK')* be the Levenshtein distance between *T'* and *TK'*. The phonetic similarity can be calculated by equation (1), where length(s) represents the number of characters in string s. EKTP candidates with the condition $sim(E,K) > \sigma$[3], would be extracted as EKTPs.

$$sim(E,K) = \frac{length(K') - LD(K',TK')}{length(K')} \qquad \textbf{(1)}$$

Table 1. Mapping table from consonants to their phonetic code: consonants with '*' means that they are used as the last consonants in Korean syllables, while others are used as the first consonants in Korean syllables.

Consonants	Phonetic code
'g', 'g*', 'gg', 'k'	1
'n', 'n*', 'ng*'	2
'd', 'dd', 't', 't*', 'ch'	3
'l', 'l*'	4
'm', 'm*'	5
'b', 'b*', 'bb', 'p', 'h'	6
's', 'ss', 'j', 'jj'	7

3 Experiments

We prepare two data sets for evaluating our proposed method. One is for evaluating precision rate (precision set). The other is for evaluating recall rate (recall set). The precision set is entries of bilingual domain-specific dictionaries, which contain 1,400,000 English-Korean translation pairs. The recall set contains manually constructed EKTPs – about 7,000 entries (Nam, 1997). The results are evaluated with precision, recall, F-value. Precision means that the proportion of the number of relevant EKTPs, to the total number of extracted EKTPs. Recall means that the proportion of the number of extracted EKTPs, to the total number of manually constructed EK TPs. F-value is a combined measure of precision and recall rate (Salton, 1983).

3.1 Evaluation Results

Table 2 shows the result of our method. The result indicates that "Comparing phonetic similarity" effectively excludes errors produced by "Extracting EKTP candidates" without great

[2] The KODEX code is used for a KODEX algorithm. The KODEX algorithm, like SOUNDEX algorithm, is one that finds the phonetically similar words. The reader can regard the KODEX algorithm as the Korean version of the SOUNDEX algorithm. The difference may be the code conversion table, say KODEX code and SOUNDEX code. In this paper, we will compare the algorithm with our third step in the experiment. Note that our algorithm does not use KODEX algorithm but just uses the KODEX code. The KODEX algorithm determines that two strings are same when phonetic codes of consonants in the two strings are same. Moreover the KODEX algorithm does not care vowels.

[3] In this paper, we set the threshold σ as 0.5. In the experiment, we will show impact of the threshold on an EKTP acquisition task.

loss of recall rate. "Comparing phonetic similarity" improves the precision rate about 11.5% with 1.5% loss of the recall rate. Totally, the performance of F-value is improved about 5.56%. Though the recall rate is relatively low, the precision rate is very high nearly 99%.

We acquire about 20,000 EKTP candidates from the precision set and about 5,300 EKTP candidates from the recall set. Low recall rate is caused by the constraint – all strings of Korean entries in E-K translation pairs of the dictionary should be transliterations. Actually, the performance of recognizing transliterations is relatively high – about 92%~98% precision and about 95% recall. However, the rigid constraint makes it difficult to extract EKTP candidate effectively, though there is only one syllable tagging error in the first step. This is the reason why the recall rate is relatively low.

Table 2. Evaluation results of the proposed method

	Precision	Recall	F-measure
Step 1	88.64%	73.72%	81.18%
Step 3	98.76% (+11.42%)	72.64% (-1.47%)	85.70% (+5.56%)

Table 3. Examples of EKTP candidates

English words	Korean words
chiral	'ki-ral'
chromatogram	'keu-lo-ma-to-geu-raem'
chromatograph	*'keu-lo-ma-to'*
conterminous grafting	'kon-teo-mi-neo-seu geu-la-peu-ting'
degrease	*'tal-geu-li-seu'*
Diad	'di-a-deu'

Table 3 shows some EKTP candidates produced by "Extracting EKTP candidates" step. In the example, the underlined pairs are EKTP candidates. In the EKTP candidate <*chromatograph*, 'keu-lo-ma-to'>, 'keu-lo-ma-to' is a transliteration for 'chromato' rather than 'chromatograph'. Note that the relevant transliteration for *chromatograph* is 'keu-lo-ma-to-geu-la-peu'. <*degrease*, 'tal-geu-li-seu'>, where 'tal' is a pure Korean word, is also a wrong EKTP candidate . Because the first step wrongly tags 'tal' as 'F', 'tal-geu-li-seu' is determined as a transliteration. Note that 'di-geu-li-seu' is the correct transliteration for *degrease*. By comparing phonetic similarity, the noisy EKTP candidates are filtered out.

Table 4 Examples of transliteration equivalence

English word	Korean word
amidase	'a-mi-de-i-seu', 'a-mi-da-a-je', 'a-mi-da-je'
cytophore	'sa-i-to-po-eo', 'sa-i-to-po'
desmolase	'de-seu-mol-le-i-seu', 'de-seu-mol-la-a-je', 'de-seu-mol-la-je'
ferredoxin	'pe-le-dog-sin', 'pe-lo-dog-sin'
ferrite	'pe-la-i-teu', 'pe-li-teu'
zymogen	'ji-mo-gen', 'ja-i-mo-jen'

Our algorithm extracts about 52,000 transliteration pairs and about 42,000 transliteration equivalences from the precision set. Table 4 shows some examples of transliteration equivalence.

4.2 Comparing with Previous Works.

Table 5. Comparison of ours and the previous method

	Precision	Recall	F-value
A	98.40%	68.48%	83.44%
B	98.10%	70.49%	84.29%
C	99.59%	60.92%	80.26%
D	98.76%	72.64%	85.70%

In this section, we compare our method with the previous works. In this experiment, we only compare the third step of our method with that of others, because other methods mainly focus on comparing phonetic similarity. Table 6 shows the result of the comparison. In the table, A, B, C, and D represent Levenshtein distance (Levenshtein, 1965), Dice coefficient (Tsujii, 2002), KODEX algorithm (Kang, 2001) and our proposed method, respectively. The result shows that our method outperforms others, especially for recall rate – 6.07% improvement for A, 3.05% improvement for B, 19.24% for C.

4.3 Evaluation according to threshold

In this section, we will describe the effect of the threshold value used in the third step on the performance. Table 6 shows the result. In the result, we find that threshold 0.5 is the optimal value, which produces the best performance. We assume that higher threshold tends to show lower recall rate and higher precision rate, while lower threshold tends to show higher recall rate and lower precision rate. In the result, we find that the assumption on the higher threshold is right – the performance sharply decreases from threshold 0.7 to 1.0. The high threshold value forces the algorithms to exclude EKTP candidates, which Korean words and transliterations of English words are not exactly equivalent. Therefore the recall rate sharply decreases.

Table 6. Performance according to threshold.

Threshold	Precision	Recall	F-value
0	88.64%	73.72%	81.18%
0.1	96.13%	73.61%	84.87%
0.2	96.73%	73.57%	85.15%
0.3	97.56%	73.34%	85.45%
0.4	98.16%	73.06%	85.61%
0.5	98.76%	72.64%	85.70%
0.6	99.29%	70.47%	84.88%
0.7	99.59%	65.62%	82.60%
0.8	99.67%	56.93%	78.30%
0.9	99.85%	37.63%	68.74%
1	99.84%	29.99%	64.91%

On the contrary, the assumption on the lower threshold value does not agree with our result – the precision of threshold value 0 is about 88%. The threshold value 0 means that no EKTPs are filtered out in the third step. This means that the threshold value 0 is the performance of the first step. The result indicates that the performance of the first step is very important to improve that of the third step.

Conclusion

This paper has described a method for English-Korean transliteration pair acquisition. Our method extracts EKTP candidates and then finds out the relevant EKTP by comparing phonetic similarity. Evaluation results show that step 1 extracts EKTPs with relatively high precision and step 3 can improve precision rate without great loss of recall rate. Moreover, we show that our method outperforms the previous ones.

In the future work, we will devise an algorithm for extracting EKTPs from bilingual corpora. We expect that our method can be ported to corpus-based EKTP extraction without great changes. The transliteration equivalence recognized by our algorithm makes it possible to enrich Korean or English-Korean domain specific thesauri.

Acknowledgements

This work was supported by the Korea Ministry of Science and Technology, the Korea Ministry of Commerce, Industry and Energy, and the Ko-rea Science and Engineering Foundation (KOSEF)

References

Brill E., Gary Kacmarcik, Chris Brockett, (2001): Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs. NLPRS 2001: 393-399

Collier, N., Kumano, A. and Hirakawa, H. Acquisition of English-Japanese proper nouns from noisy-parallel newswire articles using KATAKANA matching. In Proceedings of the Natural Language Processing Pacific Rim Symposium 1997. 1997, pp. 309 - 314.

Kang B. J., (2001), A Resolution of Word Mismatch Problem Caused by Foreign Word Transliterations and English Words in Korean Information Retrieval, PhD thesis, Computer Dept., Korea Advanced Institute of Science and Technology

Knight, K. and J. Graehl, (1997). "Machine Transliteration". In Proceedings. of the 35th

Annual Meetings of the Association for Computational Linguistics (ACL) Madrid, Spain.

Levenshtein V. I.. (1965) Binary codes capable of correcting deletions, insertions and rever-sals. Doklady Akademii Nauk SSSR 163(4) p845-848,

Nam Y. S., (1997), the foreign word dictionary Sung-An-Dang Publisher

Odel, M. K. And Russell, R.C., (1918), U.S. Patent No. 1261167(1918) and 1435663(1922)

Oh, J.H, and Key-Sun Choi, (2003), A statistical model for Automatic Extraction of Korean Transliterated Foreign words, International Journal of Computer Processing of Oriental Lan-guages (IJCPOL), Vol. 16, No.1.

Oh J.H., and Key-Sun Choi, (2005), Machine Learning Based English-to-Korean Translitera-tion Using Grapheme And Phoneme Information, Journal of IEICE transactions on Informa-tion Systems, Vol 88, No. 7

Salton, G. and McGill, M. (1983), Introduction to Modern Information Retrieval, New-York: McGraw-Hill

Tsuji, K. (2002), Automatic Extraction of Translational Japanese-KATAKANA and Eng-lish Word Pairs from Bilingual Corpora, International Journal of Computer Processing of Oriental Languages, vol.15, no.3, p.261-279