Extracting English-Korean Transliteration Equivalence from Domain-Specific Dictionaries

Jong-Hoon Oh and Hitoshi Isahara
Computational Linguistics Group,
National Institute of Information and Communications Technology (NICT)
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{rovellia, isahara}@nict.go.jp

Abstract

Automatic translation knowledge acquisition or automatic bilingual dictionary construction has become an important first step for natural language applications such as machine translation and cross-language information retrieval. Transliterations are 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 composed of one foreign word and its possible transliterations. Transliterations are one of the main sources of the out-of-vocabulary (OOV) problem, because transliteration is a productive process. Many Korean domain-specific terms are composed of transliterations. Therefore, translation knowledge on transliteration equivalence is important for natural language applications to process domain-specific texts. In this paper, we propose an algorithm recognizing transliteration equivalence or transliteration pairs in domain-specific dictionaries using machine transliteration. Our method shows about 99% precision and 73% recall rate.

1 Introduction

Automatic translation knowledge acquisition or automatic bilingual dictionary construction has become an important first step for natural language applications such as machine translation and cross-language information retrieval. *Transliteration* can be defined as phonetic translation. Transliterations in languages using non-roman alphabets such as Korean, Chinese, and Japanese, are conventionally used to represent foreign loan words that have been imported to their languages. For example, English word data is transliterated into Korean as 'de-i-teo'[1]. Transliteration is usually used to translate proper names and technical terms. They are one of the main sources of the out-of-vocabulary (OOV) problem, because transliteration is a productive process. Thus, dictionary look-up is an impractical way when we handle transliterations and their original word. *Transliteration equivalence* refers to a set composed of an original foreign word and its possible transliterations. For example, a set composed of the 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 standard transliteration. However, it is difficult to discriminate the standard transliteration from its variations especially for coined terms. Therefore we do not distinguish between the two in this paper.

Many Korean domain-specific terms are composed of transliterations [1]. For example, Korean biological terms, 'a-mil-la-a-je' and 'a-de-nil peb-ti-da-a-je' are Korean transliterations of *amylase* and *adenyl peptidase*, respectively. Therefore, translation knowledge on transliteration equivalence is important for natural language applications to process domain-specific texts. In this paper, we propose an algorithm recognizing transliteration equivalence or transliteration pairs in domain-specific dictionaries using machine transliteration algorithm. The goal of our method is to find transliteration equivalence from English-Korean translation pairs, which are entries of domain-specific dictionaries. For transliteration pair acquisition, we need to phonetically convert words in one language to that in the other language to phonetically compare one with the other. Machine transliteration can serve as a component for the phonetic conversion by offering machine-generated transliterations.

This paper organized as follows. In Section 2, we describe the previous works. Section 3 shows details of our method. Section 4 deals with experiments. Conclusion and future works are drawn in Section 5.

2 Previous works

Transliteration pair acquisition has been received significant attention from many researchers especially between English and Japanese [2, 3, 4]. Collier *et al.* [3] aimed at extracting transliteration pairs for proper names. His method was composed of two steps. First, he tried to extract transliteration pair candidates where the first letter of English words were in the upper case and Japanese words were written in *katakana*. Second, transliteration pairs were recognized through NPT (Nearest Phonetic Transliteration) transcription and string similarity. Japanese terms in candidates were transformed into NPT tran-

[1] In this paper, Korean transliterations are represented in their Romanized form with quotation marks (''). Note that '-' will be used as a syllable boundary in the Korean transcription.

scription, then transliteration pairs were found by comparing similarity between English word and NPT transcription. Tsuji [4] proposed an algorithm 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 transliteration rules observed in the training corpora. Brill *et al.* [2] proposed a statistical transliteration pair acquisition model based on 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 differences 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 representing loan words or transliterations, called *katakana*. We can easily find Japanese transliterations just by investigating strings written in *katakana*. However, it is hard to recognize Korean transliterations in Korean texts, because pure Korean words and transliterations share the same character set. Therefore, an algorithm is necessary to detect and recognize Korean transliterations in Korean texts. Second one is a way of phonetic conversion. The phonetic conversion procedure in the previous works, is back-transliteration (Japanese to English), while ours is transliteration (English to Korean) in the context of Knight & Graehl [5]. Because of the information losing aspect of the transliteration process, the back-transliteration is harder than transliteration [5]. Therefore, machine transliteration can generates phonetically equivalent strings more accurately than back-transliteration.

3 Method

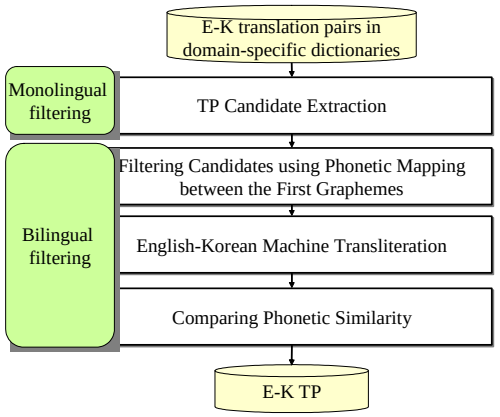

Figure 1: The overall system architecture

Figure 1 shows the overall system architecture of our proposed method composed of four steps. First, the system tries to extract transliteration pair (TP) candidates by recognizing Korean transliterations. The system filters out E-K translation pairs, where Korean parts contains no transliteration. Once one or more than one Korean transliterations are detected, the system make TP candidates with all possible combinations between English word and Korean transliteration. Second, the system filters out candidates by using phonetic mapping information between the first graphemes in English and Korean. The system regards TP candidates as relevant ones when there is phonetic mapping relation between the first graphemes in each language. Third, the system automatically transliterates English word in candidates into Korean. Finally, the system recognizes TPs through phonetic similarity between Korean word and machine generated transliteration. Because the first step makes use of Korean transliterations to find candidates, we call it a monolingual filtering step; while the second and fourth step are called a bilingual filtering step because they use both English and Korean.

3.1 TP Candidate Extraction

In this step, we use syllable-tagging to detect Korean transliterations proposed by [1]. For a given word phrase S, the syllable-tagging algorithm tags each syllable in the word phrase as either 'F' or 'K' maximizing $P(S|T)P(T)$ in formula (1). Note that a syllable with tag 'F' means that it is part of a transliteration; while that with tag 'K' means that it 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-je' (*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 + je/F", by the algorithm. A series of 'F' tags makes it possible to detect and extract transliterations. If there is a series of 'F' tags in the syllable tagged result, we can determine that a given word phrase contains transliterations and the words corresponding to the series of 'F' tags can be extracted as a Korean transliteration. However, we apply a strict constraint that all syllables in a Korean word phrase should be tagged as 'F' in order to reduce errors caused by this step. Once Korean transliterations are detected, we make a list of TP candidates using combinations between English word and detected Korean transliteration. For example, we can extract TP candidates, <*adenyl*, 'a-de-nil'>, <*adenyl*, 'peb-ti-da-a-je'>, <*peptidase*, 'a-de-nil'>, and <*peptidase*, 'peb-ti-da-a-je'> from the translation pair <*adenyl peptidase*, 'a-de-nil pep-ti-da-a-je'>.

$$\phi(S) = argmax_T P(S|T)P(T) \quad (1)$$

$$P(S|T) = \prod_{i=1}^{n} p(s_i|t_i, t_{i-1}, s_{i-1})$$

$$P(T) = p(t_1|t_0) \times \prod_{i=2}^{n} p(t_i|t_{i-1}, t_{i-2})$$

Table 1: KODEX Code

Consonants	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

3.2 Filtering Candidates using Phonetic Mapping between the First Graphemes

In this step, we filter out TP candidates using phonetic mapping between the first graphemes. The first grapheme between English word and its Korean transliteration shows a canonical phonetic mapping relation. For example, the first grapheme *b* in English words is usually transliterated into Korean graphemes 'b'. We extract the phonetic mapping relation between the first grapheme, which appears more than ten times in an E-K transliteration dictionary (Note it covers 99% of the transliteration dictionary). Let M_{e_i} be the set of all possible Korean graphemes transliterated from e_i. The filtering in this step can be described as formula (2), where e_i and k_i represent the i^{th} grapheme in each language, $E = e_1, \cdots, e_n$ and $K = k_1, \cdots, k_m$.

$$Sim_{fg}(E,K) = \begin{cases} 1, & k_1 \in M_{e_1} \\ 0, & otherwise \end{cases} \quad (2)$$

3.3 English-Korean Machine Transliteration

In this step, we use a correspondence based transliteration model for English-to-Korean machine transliteration [6]. The transliteration model transforms English words into Korean transliterations using correspondence between English grapheme and English phoneme. All English words in TP candidates are transliterated into Korean, in this step.

3.4 Comparing Phonetic Similarity

Let K and E be Korean words and English words in TP candidates, respectively, and TK be a transliteration of E produced by the machine transliteration step. Relevant TPs are selected by comparing phonetic similarity between K and TK. For the comparison, consonants in K and TK are converted into KODEX code (see Table 1 [7]), which contains seven Korean consonant groups resulting in similar pronunciations.

Let *T'* and *TK'* be the strings when we apply KODEX code to T and TK, respectively. Then the phonetic similarity function can be represented as formula (3) where *length(s)* represents the number of graphemes in string s and $LD(T', TK')$ is the Levenshtein distance between *T'* and *TK'*. The similarity function is based on Levenshtein Distance (LD). The distance means the number of deletions, insertions, or substitutions required to transform source string into target string [8]. Finally, we can recognize E-K TP with the condition $Sim(E,K) \geq \sigma$ as described in formula (4).

$$Sim_p(E,K) = Sim_p(TK,K) = Sim_p(TK',K') = \frac{length(K') - LD(K',TK')}{length(K')} \quad (3)$$

$$Sim(E,K) = Sim_{fg}(E,K) \times Sim_p(E,K) \geq \sigma \quad (4)$$

Table 2: Performance of monolingual filtering

Precision	Recall	F-value
88.64%	73.72%	81.18%

Table 3: Errors produced by monolingual filtering

English	Korean
chromatograph	'keu-lo-ma-to'
degreasing	'tal-geu-li-seu'
ermine	'eo-min'

4 Experiments

4.1 Experimental Setup

We prepare two data sets for evaluating our system. 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 about 1,400,000 English-Korean translation pairs. The recall set is an E-K transliteration dictionary containing 7,000 entries [9]. We evaluate the performance of monolingual filtering (step 1) and that of the whole system. The results are evaluated with precision, recall, and F-value. Precision means that the proportion of the number of relevant TPs to the total number of extracted TPs. Recall means that the proportion of the number of extracted TPs to the total number of TPs in the recall set. F-value is defined as formula (5).

$$F - value = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

4.2 Monolingual Filtering and Bilingual Filtering

Table 2 shows the performance of monolingual filtering. We acquire about 20,000 E-K translation pairs containing Korean transliterations (about 100,000 TP candidates) from the precision set and about 5,300 TP candidates from the recall set through monolingual filtering. In the result, recall rate is relatively low. It is due to errors in syllable tagging and the strict constraints – all strings of a Korean word phrase should be transliterations. The performance of the syllable tagging is relatively high – about 92%∼98% precision and about 95% recall in a syllable level. However, we can not recognize TPs even if there is only one syllable tagging error.

Table 4: Performance of Bilingual filtering

	Precision	Recall	F-value
Mono	88.64%	73.72%	81.18%
Mono+Bi	98.76%	72.64%	85.70%

Table 5: Performance Comparison

	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%

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%

Table 3 shows such errors. Note that the underlined grapheme in Korean column is not a part of the transliteration. In the case of *chromatograph*, its Korean part 'keu-lo-ma-to' is a transliteration. But 'keu-lo-ma-to' is a counterpart not for *chromatograph* but for *chromato*. In the case of *degreasing*, the Korean word phrase is not transliteration because of the syllable 'tal'. However, the syllable 'tal' is wrongly tagged as 'F' and the wrong TP is extracted. We expect that bilingual filtering can check this type of error by calculating phonetic similarity. We can not extract *ermine* and its Korean transliteration 'eomin' as a TP candidate because of the syllable tagging error (it was tagged as "eo/K+min/K").

Table 4 shows the performance of bilingual filtering. The result indicates that bilingual filtering effectively excludes errors produced by monolingual filtering without great loss of recall rate. Bilingual filtering improves precision rate about 11.5% with 1.5% loss of recall rate. Totally, the performance of F-value is improved about 5.56%.

4.3 Comparing to previous works

In this section, we compare our method and the previous works. In this experiment, we compare only the fourth step of our method with others, because other method mainly focuses on phonetic similarity. Table 5 shows the result of this experiment. In the table, A, B, C, and D represent Levenshetein Distance [8], Dice coefficient [4], KODEX algorithm [7] and our proposed method, respectively. The result shows that our method outperforms other methods, especially on recall rate – 6.07% for A, 3.05% for B, 19.24% for C.

4.4 Evaluation according to threshold

In the result, we find that threshold 0.5 is the optimal value, which produces the best performance. We expect 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 Table 6, the performance for high threshold value agrees with our expectation but that for low threshold value does not – the precision of threshold value 0 is about 88%. The threshold value 0 means that no TP candidates are filtered out in the fourth 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 the overall system performance.

5 Conclusion

This paper has described a method for English-Korean transliteration pair acquisition. Our method use two kinds of filtering method, say monolingual filtering and bilingual filtering. Evaluation results show that monolingual filtering extracts E-K TPs with relatively high precision and bilingual filtering can improve precision rate without great loss of recall rate. Moreover, we show that our method outperforms the previous ones. In the future works, we will devise an algorithm for extracting E-K TPs from bilingual corpora. We expect that our method can be ported to corpus-based E-K TP acquisition without great changes.

References

[1] Jong-Hoon Oh and Key-Sun Choi. A statistical model for automatic extraction of korean transliterated foreign words. *International Journal of Computer Processing of Oriental Languages (IJCPOL)*, 16 (1), 2003.

[2] E. Brill, G. Kacmarcik, and C. Brockett. Automatically harvesting katakana-english term pairs from search engine query logs. In *Proc. of the Natural Language Processing Pacific Rim Symposium 2001*, pages 393–399, 2001.

[3] N. Collier, A. Kumano, and H. Hirakawa. Acquisition of english-japanese proper nouns from noisy-parallel newswire articles using katakana matching. In *Proc. of the Natural Language Processing Pacific Rim Symposium 1997*, pages 309–314, 1997.

[4] K. Tsujii. Automatic extraction of translational japanese-katakana and english word pairs from bilingual corpora. *International Journal of Computer Processing of Oriental Languages*, 15(3):261–279, 2002.

[5] K. Knight and J. Graehl. Machine transliteration. In *Proceedings of the 35th Annual Meetings of the Association for Computational Linguistics*, 1997.

[6] J. H. Oh and K. S. Choi. An ensemble of grapheme and phoneme for machine transliteration. In *the Proceedings of IJCNLP*, 2005.

[7] B. J. Kang. *A resolution of word mismatch problem caused by foreign word transliterations and English words in Korean information retrieval.* PhD thesis, Computer Science Dept., KAIST, 2001.

[8] V. I. Levenshtein. Binary codes capable of correcting, deletions, insertions and reversals. *Soviet Phys. Dokl*, 10:707–710, 1966.

[9] Y. S. Nam. *Foreign dictionary.* Sung An Dang, 1997.