Two Approaches for The Resolution of Word Mismatch Problem Caused by English Words and Foreign Words in Korean Information Retrieval

Byung-Ju Kang and Key-Sun Choi

Korea Advanced Institute of Science and Technology
Department of Computer Science
Advanced Information Technology Research Center
Korea Terminology Research Center for Language and Knowledge Engineering
Daejeon, Korea
Email: bjkang@world.kaist.ac.kr, kschoi@world.kaist.ac.kr

Abstract

In Korean text, recently, the use of English words with or without phonetic translation is growing at high speed. To make matters worse the Korean transliterations of an English word may be very various. The mixed use of English words and their various transliterations may cause severe word mismatch problem in Korean information retrieval. There can be two possible approaches, transliteration and back-transliteration method, to tackle the problem. We argue that our newly proposed transliteration approach is more advantageous for the resolution of the word mismatch problem than the previously proposed back-transliteration approach. Our information retrieval experiment results support this argument.

Keywords: information retrieval, Korean information retrieval, transliteration, back-transliteration, word mismatch problem.

1 Introduction

In Korean text, recently, the use of foreign words, which are mostly transliterations of English words, is growing at high speed. This is mainly due to the World Wide Web and the Internet that enable the instant access of new information at the global scale. Korean transliteration of an English word may be very various. For example, English word 'digital' may be variously transliterated in Korean as '디지털 (ticithel)', '디지탈 (ticithal)', and '디지틀 (ticithul)', etc, even though '디지털 (ticithel)' is preferred as a standard form. This is because an English phoneme can only be ambiguously mapped to more than one Korean phoneme due to their radically different phonologies. Moreover, writers often use English words in their original forms without transliterating them. These mixed use of various transliterations together with their origin English word cause severe word mismatch problem in information retrieval [1, 2]. When user query and document text use different transliteration each other, simple word matching cannot retrieve the document. When user query uses Korean transliteration and document contains English word or vice versa, simple word matching also fails.

In order to resolve the word mismatch problem, it is necessary to find equivalence classes among English words and their various Korean transliterations. However constructing the equivalence classes is not easy due to the inherent difficulties of the problem. There can be two possible approaches to tackle the problem. One approach is to transform, i.e. back-transliterate, foreign words into their origin words (English) and use the English words as canonical forms for indexing and querying [1]. The other approach, which is originally proposed in this paper, is to transliterate English words into Korean and construct equivalence classes among foreign words by measuring the phonetic similarities among them. The back-transliteration approach appears to be more convincing since English word is unique and its Korean transliteration is multiple [1]. However the back-transliteration approach has more difficulties in its actual implementation than the transliteration approach has. First, back-transliteration is inherently more difficult than transliteration [3]. Transliteration is an irreversible process where the phonetic information of the origin word is lost. Therefore, without recovering this information, perfect back-transliteration is impossible. So, generally post-processing of approximate dictionary matching is performed to find right English word [1, 2]. But the approximate matching accumulates another errors by looking for wrong words. Secondly, typically in Korean text, there exist much more foreign

Proceedings of the 5th International Workshop Information Retrieval with Asian Languages

Copyright ACM 1-58113-300-6/00/009 ... $5.00

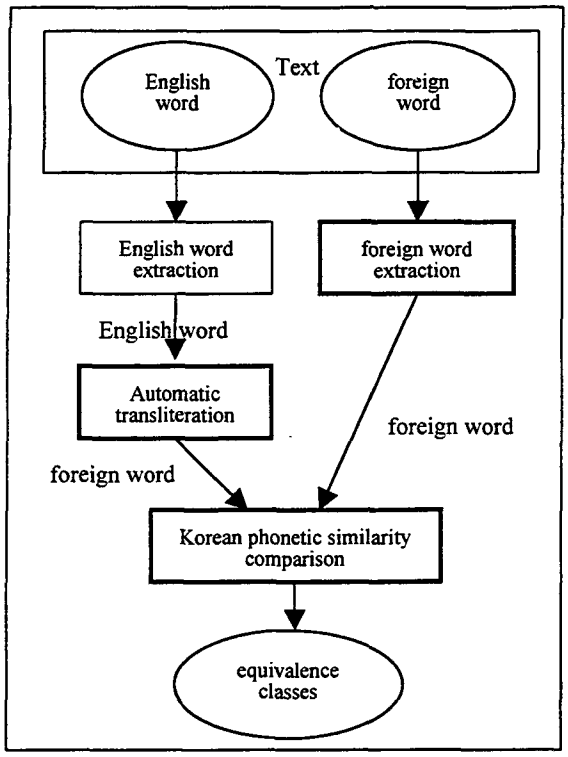

Fig. 1. The transliteration approach

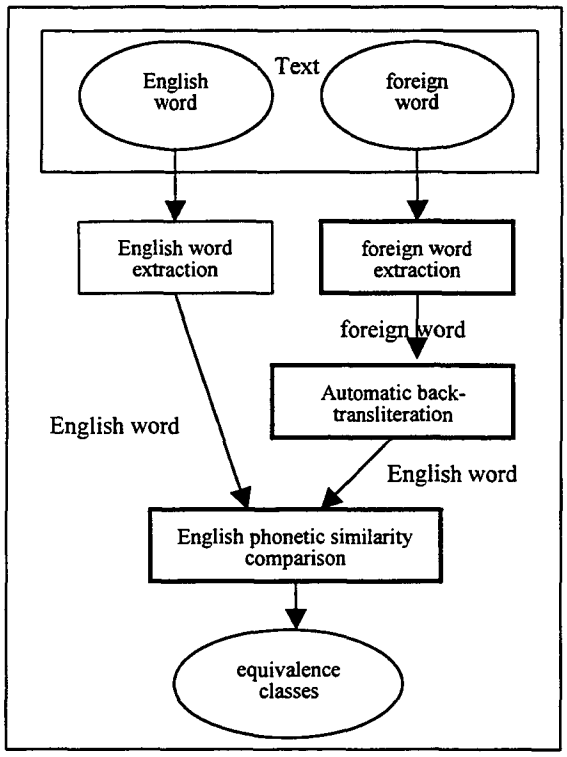

Fig.2. The back-transliteration approach

words than English words. Therefore, in the case of back-transliteration of more foreign words, more errors may be generated in total than in the case of transliteration of fewer English words. Third, English multi-word problem is more difficult to be handled in back-transliteration than in transliteration. In most cases English multi-word is transliterated into a single Korean compound word. So, in back-transliteration, the foreign compound word need to be segmented into component foreign words so that each component word may be converted to appropriate English word. The problem is that compound word segmentation is very ambiguous especially when unknown words are involved. On the other hand, in the case of transliteration each part of an English multi-word may be independently transliterated and then simply concatenating them in most cases results in the right transliteration of the multi-word. For example, given an English multi-word "Web server", 'Web' and 'server' is transliterated respectively into '웹 (wep)' and '서버 (sobo)' and concatenating them results in correct transliteration "웹서버 (wepsobo)". From these three reasons, we argue that the transliteration approach is more effective than the back-transliteration approach for the resolution of the word mismatch problem caused by English word and its various Korean transliterations.

In this paper we implement the two approaches, transliteration and back-transliteration approach, and compare their relative effectiveness in Korean information retrieval. In the transliteration approach (Fig. 1), first, foreign words and English words are extracted and then English words are transliterated into Korean phonetic equivalents. Lastly, by measuring phonetic similarities between foreign words, equivalence classes are constructed. In the back-transliteration approach (Fig. 2), on the other hand, first foreign words and English words are extracted and then foreign words are back-transliterated into their origin English word. Next, by measuring phonetic similarities between English strings, equivalence classes are constructed. The retrieval effectiveness of the resolution of the word mismatch problem is not easy to be accurately measured since there does not exist a test collection that is made appropriate for revealing the impact on retrieval effectiveness. So, we focus on revealing the performance difference between the two approaches.

2 Foreign Word Extraction

In both transliteration approach and back-transliteration approach the first task to be done is to extract foreign words. However foreign word extraction is not trivial. This is because Asian word segmentation is required due to the agglutinative nature of Korean language and most of foreign words are unknown words. In Korean not all words in a sentence are put together in a single long string like Chinese, Japanese, etc, but some of the words are written without space between them. In Korean one or more functional words may be attached to a content word (noun, verb, adjective, etc). Moreover, nouns are relatively freely joined together to form a compound noun. The text segment that is delimited by space is called *eojeol*. For example, an eojeol "분산데이터베이스시스템은 (*pwun-san-te-yi-the-pe-yi-su-si-su-theym-un*)", which consists of

three nouns and a functional word, should be segmented as follows:

분산 (*pwun-san*, distribution) / 데이터베이스 (*te-yi-the-pe-yi-su*, database) / 시스템 (*si-su-theym*, system) + 은 (*un*, func. word, subject marker)

where '분산 (*pwun-san*)' is a Korean noun and '데이터베이스 (*te-yi-the-pe-yi-su*, database)' and '시스템 (*si-su-theym*, system)' are foreign words.

We developed a new effective method of foreign word extraction through word segmentation [4]. Our method mainly consists of three parts: functional word detachment, compound noun segmentation, and foreign word detection. First, functional words are detached if there is any. In the above example, '은 (un)' is detached. Second, the remained noun sequence, "분산데이터베이스시스템", is segmented into the component nouns like "분산/데이터베이스/시스템". Lastly, foreign words, '데이터베이스 (database)' and '시스템 (system)' are detected.

Generally functional words are stripped off using a dictionary of functional words and a dictionary of nouns. However, especially when unknown words are involved, there are ambiguities where the separation should happen. Compound noun segmentation also has similar separation decision ambiguities. In order to reduce these ambiguities, we use unknown word information automatically compiled from the target corpus. Repeatedly occurring strings or substrings are good candidates of word. High frequency unknown words are relatively reliably identified but low frequency unknown words cannot be identified with enough confidence. So, in order to utilize low frequency unknown word information as well as high frequency unknown word information, we use different segmentation algorithm depending on the frequency of unknown words [4].

Once segmentation is done, the next step is to determine which is foreign word. In Korean it is possible to relatively accurately detect foreign words since the syllable sequences in transliterated foreign words are usually very rare in pure Korean words. The differences in syllable pattern stems from the drastically different phonetic system of Korean and English. So, statistical methods utilizing the differences in syllable unigram or bigram patterns between pure Korean word and foreign word have been developed [5, 6]. However one of the difficulties in the purely statistical methods is that it is very difficult to distinguish foreign word syllables and functional word syllables. This is because most of the syllables used in functional words are also frequently used in foreign words. To alleviate this problem, we modified Oh & Choi's HMM-based syllable tagging model [6], where each syllable in a word is tagged with Korean word syllable tag and foreign word syllable tag, such that the probability of the tag sequence is maximized. Specifically we introduced one more tag for functional word syllable and used more sophisticated eojeol model [4]. For example, the previous eojeol example "분산데이터베이스시스템은" may be syllable-tagged as follows:

분	산	데	이	터	베	이	스	시	스	템	은
K	K	F	F	F	F	F	F	F	F	F	T

where K, F, and T respectively represent Korean noun syllable, foreign word syllable, and functional word syllable. Now, foreign word extraction is very straightforward. All the word segments that are obtained from the previous segmentation steps are syllable-tagged and the segments that contain more than 50% foreign syllables are decided as foreign words.

3 Automatic Transliteration and Back-transliteration

Since it is realistically impossible to list all the possible variations of Korean transliterations for every English word including names, automatic transliteration and back-transliteration is required. Automatic transliteration and back-transliteration problem is very similar respectively with text-to-speech and speech-to-text transformation problem. Various machine learning algorithms have been successfully applied to the speech/text transformation problem [7, 8, 9]. So, those machine learning methods would be also quite applicable to the transliteration and back-transliteration problem. We chose decision tree classification method for the automatic learning of the transliteration and back-transliteration rules. For any supervised machine learning, large labeled training examples must be prepared. However large phonetically aligned pairs of English word and Korean transliteration do not exist. So in the previous researches relatively small training data of hand-aligned examples were used [1] or knowledge-poor automatic alignment by unsupervised learning was tried [10]. But the automatic alignment was not accurate enough. We developed a fully automatic method that almost perfectly performs phonetic alignment between English word and Korean transliteration [11]. In the following we briefly describe our alignment algorithm and decision tree induction of Korean-English transliteration and back-transliteration rules.

3.1 Character Alignment

English/Korean character alignment is, given a source language word (English) and its phonetic equivalent in target language (Korean), to find the most phonetically probable correspondence between their characters. For example, English word 'board' is generally transliterated into '보드 (potu)'[1] in Korean and their one possible alignment is as follows:

[1] Korean characters are composed in syllable unit when they get written. The two-syllable word '보드 (potu)' may be deformed into 'ㅂㅗㄷㅡ' in character unit.

English	b	oa	r	d
	|	|	|	|
Korean	ㅂ	ㅗ	-	ㄷㅡ

Let's call the mapping unit, 'b', 'oa', 'r', 'd', 'ㅂ', 'ㅗ' 'ㄷㅡ' in the above alignment example, as PU (Pronunciation Unit) [10]. We may use decision trees for the induction of the mapping rules between English PUs and Korean PUs. Unfortunately, however, too many PUs may be produced and consequently too many decision trees need to be constructed. Moreover null PUs in the source word side makes the application of decision tree method difficult. To remedy this problem we constrain the alignment configuration. Specifically we allow only one-to-many correspondence and prohibit null PUs in the source word side.

Under these constraints the previous alignment example may be modified as follows:

English	b	o	a	r	d
	|	|	|	|	|
Korean	ㅂ	ㅗ	-	-	ㄷㅡ

On the contrary, when source word is Korean and target word is English, i.e. in back-transliteration, the alignment should be as follows:

Korean	ㅂ	ㅗ	ㄷ	ㅡ
	|	|	|	|
English	b	oar	d	-

This constrained version of character alignment makes decision tree learning more manageable and more efficient. In the case of E/K transliteration only 26 decision trees for each English alphabet need to be learned and in the case of E/K back-transliteration only 46 decision trees for each Korean alphabet need to be learned.

For the automatic character alignment, we developed an extended version of Covington's alignment algorithm [12, 13]. Covington's algorithm views an alignment as a way of stepping through two words while performing *match* or *skip* operation on each step. Thus the alignment

source	b	o	a	r	d	-
target	ㅂ	ㅗ	-	-	ㄷ	ㅡ

is produced by matching 'b' and 'ㅂ', 'o' and 'ㅗ', then skipping 'a' and 'r', matching 'd' and 'ㄷ', and lastly skipping 'ㅡ'. Null symbol '-' indicates skip at the position.

Covington's algorithm produces only one-to-one correspondence. This implies that null mapping is inevitable on both source and target word side. In order to produce one-to-many correspondences and remove null on the source word side we introduce *bind* operation. We define two kinds of bind operation: *forward bind* and *backward bind*. The following alignment example of English word 'switch' and Korean transliteration '스위치[2] (suwuichi)' pictorially represents the two bind operations.

source	ㅅ	ㅡ	>	ㅟ	>	ㅊ	<	ㅣ
target	s	-	w	i	t	c	h	-

where '>' and '<' respectively represent forward bind and backward bind at the position. 'w' is forward-binded with 'i' and together matched with 'ㅟ'. Similarly, 't' and 'h' is forward-binded and backward-binded with 'c' and collectively matched with 'ㅊ'. By introducing bind operations we can also remove null on the source side. Therefore the recurrent alignment example of 'board' and '보드 (potu)' may be represented, respectively in transliteration and back-transliteration, as follows:

source	b	o	a	r	d	<
target	ㅂ	ㅗ	-	-	ㄷ	ㅡ

source	ㅂ	ㅗ	<	<	ㄷ	ㅡ
target	b	o	a	r	d	-

We can systematically generate all the valid alignments that are possible by match, skip, bind operations and satisfy the alignment constraints. Aligning may be interpreted as finding the best alignment in the alignment search space that is composed of all the valid alignments. The algorithm does depth-first search while pruning fruitless branches early [12]. To evaluate each alignment, every match, skip, bind is assigned a penalty depending on the phonetic similarity between the English letter and Korean character under consideration. The alignment that has the least total penalty summed over all the operations is determined as the best. For example, the total penalty of the alignment of 'board' and '보드 (potu)' can be computed as follows:

English	b	o	a	r	d	<	
Korean	ㅂ	ㅗ	-	-	ㄷ	ㅡ	
operation	M	M	S	S	M	b.B	
penalty	0	10	40	60	0	200	=310

[2] When '스위치 (suwuichi)' is deformed into 'ㅅㅡㅟㅊㅣ', 'ㅇ' is dropped since it is soundless.

Human who has a little bit of bilingual phonemic knowledge can almost correctly align any English word and its Korean transliteration pair. This is because relatively simple bilingual phonemic knowledge is sufficient for the alignment task. We hope to simulate this human process. We may exploit the following two heuristics that are expected to be very effective in E/K character alignment.

H1. Consonant tends to map with consonant and vowel tends to map with vowel.

H2. There exist typical Korean transliterations for each English alphabet.

We have succeeded in aligning with high accuracy using the heuristic H1 and H2. The heuristic H1 seems to always hold except 'w'. The semi-vowel 'w' is sometimes mapped to Korean consonant even though it is usually mapped to vowels. For the heuristic H2, we can easily make a list of typical Korean transliterations for each English alphabet. Generally an English alphabet has more than one Korean character that is phonetically similar. Table 1 lists phonetically similar Korean transliterations (or characters) for several English alphabets. This simple bilingual phonemic knowledge can be coded without much effort by even non-expert. If we match English alphabet with the Korean character in the list with higher priority, in most cases we get correct alignment. To handle more complicated cases we made up of the evaluation metrics in Table 2 by extending the heuristic H1.

3.2 Learning Transliteration and Back-transliteration Rules

Once aligned English word - Korean transliteration pairs are prepared, it is very straightforward to generate large training data for the decision tree induction. For the automatic transliteration (from English to Korean) the following five mapping examples may be obtained from the constrained alignment of 'board' and '보드 (potu)'.

L3	L2	L1	(E)	R1	R2	R3		K
<	<	<	(b)	o	a	r	→	ㅂ
<	<	b	(o)	a	r	d	→	ㅗ
<	b	o	(a)	r	d	>	→	-
b	o	a	(r)	d	>	>	→	-
o	a	r	(d)	>	>	>	→	ㄷㅡ

Each example consists of 6 attribute values, left three characters and right three characters, and is labeled with the corresponding Korean transliteration. These labeled examples are classified by English alphabet and then used as training data for the learning of 26 decision trees.

Table 1. Typical Korean transliterations for several English alphabets

English alphabet	Korean transliterations
a	ㅏ ㅐ ㅑ ㅓ ㅔ ㅛ
b	ㅂ ㅂ*[3]
d	ㄷ ㄸ
o	ㅗ ㅏ
r	ㄹ ㄹ*

On the other hand, for the back-transliteration (from Korean to English) the following four mapping examples may be obtained.

L3	L2	L1	(K)	R1	R2	R3		E
<	<	<	(ㅂ)	ㅗ	ㄷ	ㅡ	→	b
<	<	ㅂ	(ㅗ)	ㄷ	ㅡ	>	→	oar
<	ㅂ	ㅗ	(ㄷ)	ㅡ	>	>	→	d
ㅂ	ㅗ	ㄷ	(ㅡ)	>	>	>	→	-

These examples are classified by Korean alphabet and then used as training data for the learning of 46 decision trees.

We use ID3-like algorithm for the learning of the decision trees. ID3 is a simple decision tree learning algorithm developed by Quinlan [14]. ID3 constructs a decision trees recursively starting at the root node. At each node an attribute is selected and tested, then examples are partitioned depending on the values of the attribute. If all the examples of a node belong to the same class, the node become a leaf and labeled with the class. If there is no more attributes remained to test, then the node become a leaf and labeled with the majority class of the examples of the node.

Once the decision trees are independently learned, the transliteration process is straightforward. Given an input English word, each English letter is mapped to Korean characters using the corresponding decision trees, then concatenating all the Korean characters produces final Korean transliteration.

[3] The consonant attached with '*' indicates that it is a syllable-final consonant. Consonants are differently pronounced depending on its position within a syllable. So, distinguishing consonants as syllable-initial, consonant preceding a vowel, and syllable-final, consonants trailing after a vowel, is more advantageous for the phonetic similarity comparison.

Table 2. Alignment evaluation metrics

operation	condition	penalty
match	similar consonant / consonant	0
	similar vowel / vowel	10
	dissimilar vowel / vowel	30
	dissimilar consonant / consonant	240
	vowel / consonant	250
skip	vowel	40
	consonant	60
bind	similar consonant / consonant	0
	similar vowel / vowel	10
	dissimilar vowel / vowel	30
	dissimilar consonant / consonant	190
	vowel / consonant	200

4 Finding Equivalent Foreign Words

In the transliteration approach, we need a method of finding equivalent foreign words, i.e. finding transliterations originated from the same English word. Actually back-transliteration is exactly the method sought for but we want to avoid back-transliteration. So we need to indirectly deduce the equivalency by measuring the phonetic similarity among foreign words.

We developed a similarity key based method, which is called Kodex, like Soundex algorithm [15] for English. Kodex maps similarly pronounced strings into identical or similar keys. Kodex compares only consonants like Soundex. Kodex works as follows:

(1) Deform a word into character unit and remove all syllable-initial 'ㅇ's if it is not the first consonant of the first syllable.

(2) Remove a syllable-final consonant if it is followed by same syllable-initial consonant.

(3) Substitute the syllable-initial consonant of the first syllable with its representative consonant if they have their representative consonants (Table 4).

(4) Replace all the consonants with their Kodex codes (Table 3) except the syllable-initial consonant of the first syllable. All vowels are removed.

(5) Remove consecutive duplicated codes that are in syllable-final and syllable-initial relationship.

For the detail rationale of each step of the above procedure, please refer to [16]. Kodex groups similarly pronounced consonants together (Table 3) based on Korean phonemic/phonetic theory and statistical observation on large foreign word data.

Table 3.Kodex consonant code table

consonants	code
ㄱ ㄱ* ㄲ ㅋ	1
ㄴ ㄴ* ㅇ ㅇ*	2
ㄷ ㄸ ㅌ ㅊ ㅅ*	3
ㄹ ㄹ*	4
ㅁ ㅁ*	5
ㅂ ㅂ* ㅃ ㅍ ㅎ	6
ㅅ ㅆ ㅈ ㅉ	7

Table 4. Representative consonants

consonant	representative consonant
ㄲ	ㄱ
ㄸ	ㄷ
ㅃ	ㅂ
ㅆ	ㅅ
ㅉ	ㅈ
ㅎ	ㅍ

String similarity measure like Damerau-Levenstein metric [17] and N-gram method [18] can be used for Korean phonetic similarity measure. In such measures, precision may be raised as high as possible by employing high threshold but low recall is unavoidable. This is because the string similarity measures fail to detect large variations in spelling that shares same sound. Determining equivalent transliterations is very simple with Kodex. Foreign words that have identical Kodex code are considered equivalent. For example, all the transliterations, '디지털 (ticithel)', '디지탈 (ticithal)', and '디지틀 (ticithul)', which are all originated from the same English word 'digital', are reduced to the same Kodex code 'D8'.

5 Experiments

For learning of the transliteration and back-transliteration rules, 7,000 English word - Korean transliteration pairs, which were selected from the foreign word dictionary of Nam [19], were prepared as raw data. 1,000 pairs out of the 7,000 word pairs were reserved for test data. The remained 6,000 word pairs were then automatically aligned by our proposed alignment algorithm.. The word accuracy about the 1,000 word test set were 51.3% and 37.2% respectively for transliteration and back-transliteration when 6,000 examples were used for training. For the more detail description of the experiment, refer to [3], [20] and [21].

For the IR experiment, KTSET 1.0 [22], which is one of the standard Korean IR test collection, was prepared. KTSET 1.0 consists of 1,000 documents and 30 queries. KTSET 1.0 were chosen since its document set is

Table 5. The impact of the resolution of the word mismatch problem on retrieval performance

recall \ experiment	Baseline	Only with more accurate foreign word extraction	Transliteration Approach	Back-transliteration Approach
0.0	0.6376	0.6293	0.6228	0.5735
0.1	0.5765	0.5756	0.5805	0.5180
0.2	0.5130	0.5267	0.5313	0.4681
0.3	0.4149	0.4773	0.4782	0.4362
0.4	0.3969	0.4547	0.4594	0.4219
0.5	0.3541	0.4152	0.4178	0.3936
0.6	0.3160	0.3721	0.3771	0.3532
0.7	0.2851	0.3133	0.3421	0.3222
0.8	0.2450	0.2671	0.2719	0.2545
0.9	0.2180	0.2360	0.2399	0.2173
1.0	0.1771	0.1985	0.2017	0.1923
Avg. precision at 11-recall point	0.3758	0.4060	0.4112	0.3773
%change	-	+8.0%	+9.4%	+0.4%
%change		-	+1.2%	-7.0%

composed of the abstract parts of technical papers from computer and information science field, so it contains relatively many English words and transliterated foreign words. As an IR search engine, SMART system [23] was used with *atc* weighting scheme for both indexing and querying.

In our experiment the back-transliteration approach was slightly differently implemented with [1]. They performed dictionary matching to find correct English word from the English string that is given as output of back-transliteration module. We don't do dictionary matching but directly compare phonetic similarity among English strings or English words. Actually, when it is an English name, dictionary matching should be avoided to prevent finding wrong words. However there is no easy way to distinguish names and ordinary nouns when they are transliterated in Korean. So, the way of our implementation should not hurt too much the performance of the back-transliteration. We used edit distance measure as the English phonetic similarity measure. Even though edit distance measure is for spelling similarity comparison, in previous researches [1, 2] it worked better than other phonetic measures such as Soundex [16] and Phonix [24].

The experiment results are shown in Table 5. You can see that the performance was greatly improved only by extracting foreign words using our foreign word extraction method. This result indicates that improvement on foreign word extraction accuracy may greatly influence the retrieval effectiveness. However the consecutive resolution of the word mismatch problem caused by English word and various Korean transliterations fails to bring any further meaningful performance improvement. This is because the IR test collection we used is not made to reveal the impact of the word mismatch problem resolution. Actually, in the query terms of KTSET 1.0, in average only 0.9 transliteration variations are used in documents. The variations are too small to reveal the impact.

In the transliteration approach, the performance was slightly improved when compared with the performance after foreign word extraction, but in the case of back-transliteration approach, on the contrary, average precision was decreased by as much as 7.0%. This result may be explained by the three reasons discussed in the introduction of this paper. In another words we may say that the back-transliteration approach is very error-sensitive due to the way of the implementation. This experiment result supports our initial belief or argument that the transliteration approach is more advantageous for the resolution of the word mismatch problem than the back-transliteration approach.

Another valuable observation is that in both transliteration and back-transliteration approach the phonetic similarity measure is very critical to the overall performance. We found that the Korean phonetic similarity measure, Kodex [16], was not good enough for the construction of accurate equivalence classes. Recall was okay but precision was too low. Hence, in order to improve performance further, we need to increase the precision of Kodex algorithm.

6 Conclusion

In this paper we presented and compared two different approaches for the resolution of the word mismatch problem caused by mixed use of English words and their various Korean transliterations in Korean text. We argued that our proposed transliteration approach is more advantages in the resolution of the word mismatch problem than the previously proposed back-transliteration approach. This argument was based on the following three observations: back-transliteration is inherently more

difficult than transliteration; typically in Korean text there exist much more foreign words than English words; English multi-word problem is more difficult to be handled in back-transliteration than in transliteration. Our IR experiment supported our argument.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

References

1. Jeong, K. S., Myaeng, S. H., Lee, J. S. and Choi, K. S. Automatic identification and back-transliteration of foreign words for information retrieval. *Information Processing and Management*, 1999, 35(4), pp. 523 - 540.
2. Lee, J. S. An English-Korean Transliteration and Retransliteration Model for Cross-lingual Information Retrieval. Ph.D. thesis, Korea Advanced Institute of Science and Technology, Dept. of Computer Science, 1998. (in Korean)
3. Kang, B. J. and Choi, K. S. Automatic transliteration and back-transliteration. Technical Report AITrc-TR-00-31-004, Advanced Information Technology Research Center, Korea Advanced Institute of Science and Technology, 2000.
4. Kang, B. J. and Choi, K. S. Effective foreign word extraction for Korean information retrieval. Technical Report AITrc-TR-00-31-003, Advanced Information Technology Research Center, Korea Advanced Institute of Science and Technology, 2000.
5. Kwon, Y. H., Jeong, K. S. and Myaeng, S. H. Foreign word identification using a statistical method for information retrieval. *In Proceedings of International Conference on Computer Processing of Oriental Languages*, Hong Kong. 1997.
6. Oh, J. H. and Choi, K. S. Automatic extraction of technical terminologies from scientific text based on hidden markov model. *In Proceedings of 11th Conference on Hangul and Korean Information Processing*. 1999. (in Korean)
7. Dietterich, T. G., Hild, H. and Bakiri, G. A Comparison of ID3 and Backpropagation for English Text-to-Speech Mapping. *Machine Learning*, 1995, 18(1), pp. 51 - 80.
8. Sejnowski, T. J. and Rosenberg, C. R. Parallel networks that learn to pronounce English text, *Complex Systems*, 1987, 1, pp. 145 – 168.
9. Stanfill. C. and Waltz, D. Toward Memory-Based Reasoning, *Communications of the ACM*, 29(12)., 1986, 29(12), pp. 1213 - 1228.
10. Lee, J. S. and Choi, K. S. English to Korean Statistical transliteration for information retrieval. *Computer Processing of Oriental Languages*, 1998, 12(1), pp. 17 - 37.
11. Kang, B. J. and Choi, K. S. Character alignment. Technical Report AITrc-TR-00-31-005, Advanced Information Technology Research Center, Korea Advanced Institute of Science and Technology, 2000.
12. Covington, M. A. An algorithm to align words for historical comparison. *Computational Linguistics*, 1996, 22(1), pp. 481 - 496.
13. Covington, M. A. Alignment of Multiple Languages for Historical Comparison. *In Proceedings of COLING-ACL '98*. 1998.
14. Quinlan, J. R. Induction of decision trees. *Machine Learning*, 1986, 1, pp. 81 - 106.
15. Odell, M. K. and R. C. Russell. U.S Patent Numbers, 1,261,167 (1918) and 1,435,663 (1922). U.S Patent Office, Washington, D.C.
16. Kang, B. J., Lee, J. S. and Choi, K. S. The phonetic similarity measure for Korean transliterations of foreign words. *Journal of Korean Information Science Society*, 1999, 26(10), pp. 1237 - 1246.
17. Wagner, R. A. and Fischer, M. J. The string-to-string correction problem. *Journal of ACM*, 1974, 22(1), pp. 168 - 178.
18. Zamora, E., Pollock, J. and Zamora, A. The use of trigram analysis for spelling error detection. *Information Processing and Management*, 1981, 17(6), pp. 305 -316.
19. Nam, Y. S. *The latest foreign word dictionary*. Sung-An-Dang Press, 1997. (in Korean).
20. Kang, B. J. and Choi, K. S. Automatic English-Korean Back-transliteration. *In Proceedings of 11th Conference on Hangul and Korean Information Processing*. 1999. (in Korean)
21. Kang, B. J. and Choi, K. S. Automatic transliteration and back-transliteration by decision tree learning. *In Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athnes, Greece. 2000.
22. Kim, J. G., Kim, Y. W. and Kim, S. H. Development of the data collection (KTSET) for Korean Information Retrieval Studies. *In Proceedings of 6th Conference on Hangul and Korean Information Processing*. 1994. (in Korean)
23. Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 1988, 24(5), pp. 8 - 36.
24. Gadd, T. PHONIX. The algorithm. *Program*, 1990, 24(4), pp. 363 - 366.