English to Korean Statistical Transliteration for Information Retrieval

JAE SUNG LEE and KEY-SUN CHOI

Department of Computer Science

Korea Advanced Institute of Science and Technology

373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, Korea

E-mail: {jslee, kschoi}@world.kaist.ac.kr

Abstract: In Korean technical documents, many English words are transliterated into Korean in various ways. Most of these words are technical terms and proper nouns that are frequently used as query terms in information retrieval systems. As the communication with foreigners increases, an automatic transliteration system is needed to find the various transliterations for the cross lingual information systems, especially for the proper nouns and technical terms which are not registered in the dictionary. In this paper, we present a language independent Statistical Transliteration Model (STM) that learns rules automatically from word-aligned pairs in order to generate transliteration variations. For the transliteration from English to Korean, we compared two methods based on STM: the pivot method and the direct method. In the pivot method, the transliteration is done in two steps: converting English words into pronunciation symbols by using the STM and then converting these symbols into Korean words by using the Korean standard conversion rule. In the direct method, English words are directly converted to Korean words by using the STM without intermediate steps. After comparing the performance of the two methods, we propose a hybrid method that is more effective to generate various transliterations and consequently to retrieve more relevant documents.

Keywords: English-Korean statistical transliteration, transliteration variations, cross lingual information retrieval, pivot method, direct method

1. Introduction

In Korean technical documents, many English words are transliterated into Korean in various forms and sometimes they are used in their original forms. Table 1 lists the approximated statistics[1] that are calculated from KT SET 2.0 test collection [20] in the areas of computer and information science. It shows that about 38.3% of word types are English words and transliterated words. And Table 2 shows the examples of variously transliterated words. These various transliterations are not negligible for information retrieval. A search with one transliteration would fail to find other variations and consequently the relevance calculation based on the frequency of the word would produce misleading results because the same words are treated as different ones. An experiment shows that the effectiveness of information retrieval increases when the various forms including English words are treated equivalently [6]. The experiment showed the necessity for the proper treatment of the various transliterations, although the process was incomplete from the automatic retrieval system's point of view, when the equivalent English words were chosen manually.

A dictionary may be used to find the transliterations and their variations, but it is not feasible because the transliterations are usually technical terms and proper nouns, not all of which can be registered in a dictionary. An automatic transliteration system is needed to find the transliterations for the novel words without manual intervention. The automatic transliteration system is definitely needed for the cross-lingual information retrieval system, too. One approach to the cross-lingual information retrieval system is a query translation method that translates the query into the language of the documents[5,25]. One of the problems of that approach is the translation of the novel words like proper nouns and new technical terms. In the case of Korean documents, those words are usually transliterated as aforementioned. As one important component of an English-to-Korean query translation, we have to make an automatic system to find the transliterated words and their variations from the English words.

There has been some research completed with regards to English-to-Korean transliteration systems. Kim [10] made an automatic transliteration system from English to Korean. The system generates English pronunciation symbols from English words by using phonological correlation rules and then the symbols are converted to Korean words according to the Korean standard conversion rule [14, 10]. However this system cannot find the variations, because the system generates only one Korean transliteration. It needs one more step to find the variations for the given Korean word; either people may replace some Korean alphabets, called "Jaso[2]", in the transliteration with confusing Jasos [24], or they may find out the variations by ignoring a few differences in Jasos among transliterations [7].

Table 1. Frequency of extracted noun words from KTSET 2.0 [20, 11]

	Korean(pure)	transliteration	English including acronyms	total
type	19,113 (61.7%)	5,394 (17.4%)	6,478 (20.9%)	30,985 (100%)
tf	358,453 (77.5%)	69,419 (15.0%)	34,873 (7.5%)	462,745 (100%)
df	225,033 (79.3%)	37,048 (13.1%)	21,655 (7.6%)	283,736 (100%)

tf = term frequency, *df* = document frequency

Table 2. Examples of various transliterations in KTSET 2.0

[1] The statistics are calculated by using a language model. Please see the procedure in section 4.2.1.

[2] The term "Jaso" is equivalent to "character" in this paper. Notice that, in some other papers, the

English word	Transliterations		
data	데이타 (teyitha) (933)	데이터 (teyithe) (563)	
digital	디지틀 (ticithul) (269)	디지탈 (ticithal) (246)	디지털 (ticithel) (4)
radio	라디오 (latio) (26)	레이디오 (leyitio) (3)	

* The number inside parenthesis is term frequency.

Two-steps of the process are combined into one step by Mettler [18], who made rules to transliterate directly from an English word to Japanese Katakana variations. He segments an English word apart phonetically and then replaces the parts with all possible corresponding Katakana substrings. The replacement procedures are described in the transliteration algorithm manually. This idea can be adopted for Korean words, but people need to make the new rules manually, because Korean is phonetically different from Japanese. Moreover, the algorithm produces too many variations. There are one or a few variations per word in most Korean transliterations as shown in Table 2.

Other approaches to English-Katakana transliteration were attempted by Kang [9] and by Collier [4]. They use an intermediate representation for both English and Katakana words to find the corresponding transliterations from bilingual corpus. Also, they made the language dependent matching or conversion rules for the intermediate representation manually.

In this paper, we propose a Statistical Transliteration Model (STM) which can generate transliterations in probability order to find the most probable ones effectively in the early stages. The STM is derived from the statistical translation model [2, 3], and we modify it to be based on basic pronunciation units that can induce the transliteration rules more effectively from word-aligned pairs. The STM is a language-independent learning system because it can learn rules automatically from word-aligned pairs of any languages. In the experiment, the STM is applied to two kinds of word pairs: an English-word and pronunciation-symbol pair and an English-word and Korean-transliteration pair.

In section 2, we survey the reason why the transliterations are various and how they are generated. To reflect the field usage, two transliteration methods are proposed: the pivot method and the direct method. The pivot method converts English words into pronunciation symbols by using the STM and then converts these symbols into Korean by using the Korean standard conversion rule. The direct method converts English words directly into Korean words by using the STM without intermediate steps.

In section 3, we will define the STM mathematically to use pronunciation units. The STM will be used as a base system for the direct method and the pivot method. In section 4, the experiments for the two methods and their results are presented. The experiments are done in three ways; first, to measure the transliteration accuracy of the methods; second, to measure how many variations the methods cover; and third, to measure the transliteration effect on the information retrieval performance. We compare two methods and propose a hybrid method to find transliteration variations more effectively. Finally, the

---

"character" is sometimes interpreted to the Korean syllable that is composed of several Jasos.

conclusion follows in section 5.

2. Background

A Korean standard transliteration rule is composed of two parts: a standard transliteration glossary and a standard transliterating method for the unregistered words in the standard glossary [14]. The transliterating method for English to Korean transliterations (hereafter E-K transliteration) is done manually in two steps using pronunciation symbols as a pivot representation; 1. English-to-pronunciation (E-P) conversion; and 2. pronunciation-to-Korean (P-K) conversion using SCR (Standard Conversion Rule) defined by the Korean government [14]. For example, the [deitə], that is a sequence of pronunciation symbols for the word "data", is retrieved from a knowledge base like a dictionary in the first step. Each pronunciation symbol is converted to Jaso according to the SCR[3]. In this case, the [d] is converted to "ㄷ", [ei] to "ㅔ이", [t] to "ㅌ" and [ə] to "ㅓ." Then, the Jasos are combined together to form a word "데이터 (teyithe) ."

The standard transliterating method can be implemented for automatic transliteration; E-P conversion may be implemented just like other automatic transcription systems [17, 19], and P-K conversion is also easily implemented using simple automata [10, 14]. However it has three problems when it is used for information retrieval systems. First, the pronunciations for some words are various; some words have alternative pronunciations and some have omissible sounds. The E-P conversion should generate the various pronunciations as well. Second, some standard transliterations, which are defined by the government [14], are not generated from the pronunciations. For example, the standard method generates "레이디오 (leyitio) " for the transliteration of "radio[reidiou]" but its standard is "라디오 (latio) ". It is because the standard glossary contains the exceptions that are not generated by the standard method if they have been used widely and over a long period of time. We evaluated the exceptions against the standard transliteration glossary [14], with consulting an English dictionary to find out the pronunciation symbols and then converting them to Korean words using the SCR. About 86% of the result is correct in *character accuracy* level and 53% in *word accuracy* level (see the evaluation definition in section 4.1.2). This means that the standard glossary and/or the transliteration usage in real documentation do not always follow the standard method and some of them are transliterated in different ways. Third, in the real usage, various transliterations are used together. Table 2 shows some examples of various transliterations retrieved from the KTSET 2.0. The standard method may not find many of them.

Lee [13] used "written word transliteration" to explain the non-standard transliteration method; it directly transliterates from written characters. For example, "ㅏ" in the word "라디오 (latio) " is

---

[3] The SCR defines mapping rules and uses context information to determine the correct corresponding Jaso; for example, if [t] is used at the end of a word, like "cat", it is converted to last Hangul consonant "

transliterated from the spelling “a” rather than the pronunciation [ei] in [reidiou]. Some variations found in Table 2 can be thought of as the written word transliterations. On the other hand, the transliterations from the words’ pronunciation was called “spoken word transliteration.” Examples of spoken word transliteration are “데이터 (teyithe) ” from “data [deitə]”, “디지틀 (ticithul) ” from “digital [didʒitl]”, and “레이디오 (leyitio) ” from “radio [reidiou]”, while “데이타 (teyitha) ”, “디지탈 (ticithal) ” and “라디오 (latio) ” are those of written word transliteration.

These two kinds of transliterations are implemented in this paper as the pivot method for “spoken word transliteration” and as the direct method for “written word transliteration.” Both methods use STM as a base model to generate various outputs in probability order. The two methods are depicted respectively in Fig. 1 and Fig. 2.

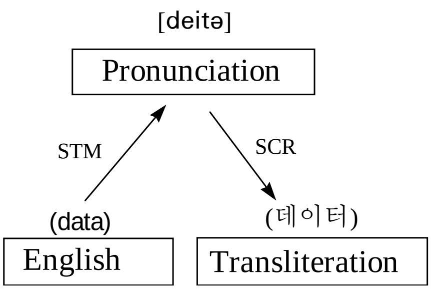

Fig. 1. The pivot method for E-K transliteration

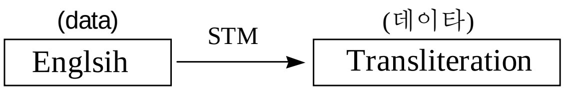

Fig. 2. The direct method for E-K transliteration

3. Statistical Transliteration Model

The transliteration problem is to find out the most probable transliteration word for a given word. Let *p(K)* be the probability of a Korean word *K*, then, for a given English word *E*, the transliteration probability of a word *K* can be written as *p(K|E)*. (In the case of the pivot method, a pronunciation symbol string, *P*, may be used instead of a Korean word *K*.) By using the Bayes’ law, we can rewrite the transliteration problem as follows [2][4]:

ㅅ” instead of “ㅌ”.

[4] Brown [2, 3] presents a statistical machine translation method for sentence translation, but we apply it to the transliteration problem.

$$\arg\max_K p(K|E)$$

$$= \arg\max_K \frac{p(K)\,p(E|K)}{p(E)} \qquad (1)$$

As *E* is given, *p(E)* is a constant. Therefore, the formula (1) can be rewritten as follows:

$$\arg\max_K p(K)\,p(E|K) \qquad (2)$$

The first factor is to estimate the language model probability and the second is to estimate the translation model probability [2, 3]. An English word, “telephone”, can be segmented into units and corresponding transliteration units are aligned to these units. If we assume that the word is segmented into *t-e-l-e-ph-o-n-e*, then one of possible alignments for the Korean transliteration “텔레폰 (theylleyphon)” can be depicted in Fig. 3.

t	e	l	e	ph	o	n	e
ㅌ	ㅔ	ㄹㄹ	ㅔ	ㅍ	ㅗ	ㄴ	null

Fig. 3. An alignment example for the word “telephone”

The translation probability for the word is represented as *p(t, e, l, e, ph, o, n, e | ㅌ, ㅔ, ㄹㄹ, ㅔ, ㅍ, ㅗ, ㄴ, null)*. The language probability is represented as *p(ㅌ, ㅔ, ㄹㄹ, ㅔ, ㅍ, ㅗ, ㄴ, null)* and it is used to find more probable word formation for the Korean word. In this example, “e” is transliterated into either “ㅔ” or “null”. If the first “e” is transliterated into “null”, then the language model probability will be zero because the Jaso sequence, “ㅌㄹㄹ” in (*ㅌ, null, ㄹㄹ, ㅔ, ㅍ, ㅗ, ㄴ, null*), never happens in proper Korean words. The zero probability will eventually reject the selection of the illegal Jaso sequence.

Alignment probabilities can be assigned by human beings but it may be different depending on each human’s judgment. We choose to use the automatic learning method not only for a fair comparison of the pivot and the direct method, but also for fast and efficient implementation. This method can be used for other language pairs too.

3.1 *PU*-based transliteration model

The characters are transliterated sequentially and some characters are transliterated as a unit. Assuming that a word is composed of these units, we define this unit as a pronunciation unit (*PU*). For convenience we use *kp* for *Korean PU* and *ep* for *English PU*. The size of *PU* is the number of characters in the unit. We use the term “*n*-size *PU*” when the size of the *PU* is *n*. We also use “*n-PU*” which means that the model uses any size *PU*'s between *1* and *n*: for example, "3-*PU*" uses *1*-size, *2*-size and *3*-size *PU*'s.

If we segment a word into *t* numbers of *PU*'s, then we can write a word, *K* (or *E*) as follows:

$$K = k_1, k_2, ...k_m \quad (3)$$
$$= kp_1, kp_2, ...kp_t \quad (4)$$

Where k_i is the *i*-th letter in *K* and kp_j is the *j*-th *PU* of *K*.

If we assume that the total number of segmentation for *K* is *N*, we can rewrite the language model as follows:

$$p(K)$$
$$= p(k_1, k_2, ...k_m) \quad (5)$$
$$= \sum^{N} p(kp_1, kp_2, ...kp_t) \quad (6)$$
$$= \sum^{N} \prod_{i=1}^{t} p(kp_i \mid kp_1, ..kp_{i-1}) \quad (7)$$
$$\approx \sum^{N} \prod_{i=1}^{t} p(kp_i \mid kp_{i-1}) \quad (8)$$

We simplified (7) to (8) by using the bigram conditional probability and a dummy initial kp_0. In the same way, if we assume that the total number of alignments for *K* and *E* is *M* and if we use bigram conditional probability, we can write the translation model as follows:

$$p(E|K)$$
$$= p(e_1, e_2, \ldots e_n | k_1, k_2, \ldots k_m) \quad (9)$$
$$= \sum^{M} p(ep_1, ep_2, \ldots ep_t | kp_1, kp_2, \ldots kp_t) \quad (10)$$
$$\approx \sum^{M} \prod_{i=1}^{t} p(ep_i | kp_i) \quad (11)$$

To estimate the probabilities of (8) and (11), we define count functions:

Definition 1: *C(x)* = the number of *x* that appears in the training data.

Definition 2: *FC(x, y)* = the number of a pair *x* and *y* appearing consecutively in the training data.

Definition 3: *AC(x, y)* = the number of aligned pair *x* and *y* in the training data.

Using the definitions, we can calculate the probability of the formula (8) and (11):

$$p(kp_j | kp_i) \approx \frac{FC(kp_i, kp_j)}{C(kp_i)} \quad (12)$$

$$p(ep_j | kp_i) \approx \frac{AC(ep_j, kp_i)}{C(kp_i)} \quad (13)$$

For calculation of (8), we have to consider N (=2^{m-1}) number of segmentations. For (11), we also have to calculate M (=$2^{m-1} \times 2^{n-1}$) number of possible alignments. This is computationally too complex and generates noisy probability. The complexity can be reduced by restricting the maximum *PU* size or by restricting the alignment distance.

3.2 Distance restriction model

The formula (13) assumes that we already know which *ep* is aligned to which *kp*. Because we do not have this information at the initial time, we use the following definition and function to estimate the initial translation probability. We utilize the position information to estimate the accurate translation probability.

Definition 4: *DC(ep, kp, d)* = the number of co-occurrence of *ep* and *kp* in the same word pair and *dist(ep, kp)* are less than or equal to *d*. The *dist(ep, kp)* is the relative distance and is calculated as follows:

$$dist(ep_j, kp_i) = \left| \, (q + \frac{s}{2}) - (p + \frac{r}{2}) \times \frac{n}{m} \, \right| \quad (14)$$

where $ep_j = e_q e_{q+1} \ldots e_{q+s-1}$, $kp_i = k_p k_{p+1} \ldots k_{p+r-1}$, *, the length of E is n and that of K is m as shown in Fig. 4.*

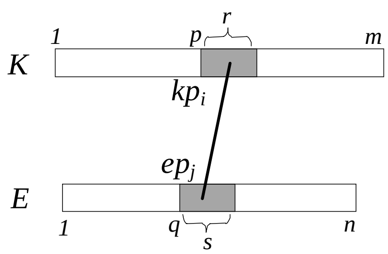

Fig. 4. *PU* distance calculation for formula (14)

Using Definition 4, we can write the formula for the initial translation probability estimation:

$$p(ep_j \mid kp_i) \approx \frac{DC(ep_j, kp_i, d)}{C(kp_i)} \quad (15)$$

3.3 Basic *PU* (1-size *PU*)

For the basic *PU*'s, we use the characters and symbols of the languages as shown in Tables 3, 4, and 5. A Korean word is composed of syllables, which can be decomposed into three components called Jaso: an initial consonant, a middle vowel, and a final consonant. The final consonants are 27 Jasos in written form, but we only use 7 Jasos for our model because they are enough to represent the sounds of the 27 final consonants [14]. Even though the shapes of the 7 final consonants are the same as those of the initial ones in Table 5, they are coded differently in the program to distinguish them from the initial ones.

Table 3. English 1-size *PU*'s (26 units)

alphabets	a – z

Table 4. Phonetic symbol 1-size *PU*'s (33 units)

consonants	p, f, b, v, t, θ, s, k, g, z, ʒ, ʃ, m, n, ŋ, l, r, d, ð, h
vowels	i, e, ɛ, æ, a, ɑ, ʌ , ɔ, o, u, ə
semi vowels	j, w
special symbol	ː

Table 5. Korean 1-size *PU*'s (45 units)

initial consonants	ㄱ, ㄲ, ㄴ, ㄷ, ㄸ, ㄹ, ㅁ, ㅂ, ㅃ, ㅅ, ㅆ, ㅈ, ㅉ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ
final consonants	ㄱ, ㄴ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ
middle vowels	ㅏ, ㅐ, ㅑ, ㅒ, ㅓ, ㅔ, ㅕ, ㅖ, ㅗ, ㅘ, ㅙ, ㅚ, ㅛ, ㅜ, ㅟ, ㅞ, ㅠ, ㅢ, ㅣ, ㅡ

3.4 Decoding

Basically the language model and the translation model are the same as the state transition probability and the output probability in the Markov model [21]. If the number of states and output symbols are large, it takes much time to decode them. In the case of the English alphabet, the number of 3-*PU* combination is $26 + 26^2 + 26^3 = 18{,}278$. The actual number of occurrences is much less but is still large. (the actual number of occurrences of the trigram is about 2,460 in the 1,500 words in our training set.)

In order to decode the model efficiently, we use the stack linguistic decoding procedure [1], which is one of heuristics to reduce the amount of computation of Viterbi algorithm. During the decoding process, a stack is used to keep only a certain number of the highest probable candidates to generate the most probable candidates in the early stages.

4. Experiment

The experiment was done in three points of view: accuracy, variation coverage and information retrieval effectiveness. We trained our model to pursue two targets, accuracy and variation, which seem to conflict with each other. This issue may be one of further research topics, but we decided to train our model first as accurately as possible and then to generate the variations with the best parameters. We think this approach will achieve the highest variation coverage, because the variations are few in most cases and they are very similar to the standard transliterations. The trained model was tested for the effectiveness of information retrieval.

4.1 Accuracy

4.1.1 Training data and testing data sets

The data for training and testing was the standard transliteration word pairs that are listed in the general area glossary of Lee [14]. This reflects some of real usage in the Korean documentation as it is described in the standard glossary rule: if some words have been widely used over a long period of time, they are accepted as the standards. Therefore, the glossary contains accepted usage in the real field. We believe that this is adequate for real usage learning even though it is not perfect.

The data contained various origin words like English, French, Russian, Italian and etc. For testing the pivot method, we chose American English pronunciation among the various ones. We eliminated some words, which were not adequate for our training, and which contained non-English alphabet or whose pronunciation could not be represented with the phonetic symbols of standard English pronunciations. But the data still contained non-English origin words, for example, “ballet”, “canzone”, and “bolsheviki”, which might degrade the learning efficiency.

The total size of the data was about 1,700 words, among which 1,500 words were randomly chosen for the training data. For the testing data, we chose 150 words from the training data and 150 words from the data that was not used for training data. We used the former data for *seen* data evaluation and the latter for *unseen* data evaluation. We call this set Test Set I.

4.1.2 Evaluation functions

Accuracy was simply measured by the percentage of the number of correct transliterations divided by the number of generated transliterations. The correct transliteration means the standard transliteration here and this measure is usually called *word accuracy*. For more accurate evaluation, we used one more function, called *character accuracy*, which measures the character edit distance between a correct word and a generated word [19].

$$word\ accuracy = \frac{number\ of\ correct\ words}{number\ of\ generated\ words} \tag{16}$$

$$character\ accuracy = \frac{L-(i+d+s)}{L} \tag{17}$$

where *L* is the length of the original string, and *i*, *d*, and *s* are the number of insertion, deletion and substitution respectively. If the dividend is negative (when $L < (i + d + s)$), we consider it as zero [19].

4.1.3 Model training for best parameters

We evaluated our model with various parameters: the *PU* size 1, 2 and 3, and *dist* = 0.3, 0.4, 0.5 and 0.6. Table 6 shows the average *character accuracy* of the *seen* and the *unseen* data when the size of Viterbi decoding stack is 10. The accuracy of 2-*PU* model was the highest for both the direct and the pivot method. And when the *dist* was 0.4, the direct performed best while the pivot performed best when *dist* was 0.3 and 0.4. The average *character accuracy* of the pivot was the same for the 0.3 and 0.4, but we chose 0.3 because *character accuracy* for the *unseen* data was higher than 0.4's *unseen* data accuracy. With the best parameters we found, we applied the training algorithm described as follows:

1. With the best parameters for each method found in the previous test, calculate the initial estimate of translation model using the formula (15) from the word aligned data.
2. Using the estimate, align the *PU*'s in each pair words to make a new character aligned data.
3. Calculate a new estimate of the translation model using the formula (13) from the character-aligned data that is made in step 2.
4. Calculate the estimate of the language model using formula (12) from the character-aligned data that is made in step 2.
5. Generate the transliterations using the estimated language model and translation model.
6. Repeat step 2 to step 5 to maximize the accuracy of transliterations.

Table 6. Average *character accuracy* for various *dist* and *PU* sizes (decoding stack size = 10)

method	Direct			Pivot		
dist \ *PU*	*1-PU*	*2-PU*	*3-PU*	*1-PU*	*2-PU*	*3-PU*
0.3	43.2%	57.1%	56.3%	54.6%	**64.4%**	61.9%
0.4	42.5%	**59.4%**	56.9%	55.2%	64.4%	62.2%
0.5	40.3%	53.6%	54.7%	55.0%	61.0%	62.4%

0.6	41.0%	53.3%	55.8%	54.2%	60.6%	63.3%

The iterative learning affected the performance of the direct method much more than that of the pivot; at the 2nd iteration, the direct increased by 8.1% and the pivot by 3.0% (see Fig. 5). Looking into the new aligned data after iteration 1, we found that many numbers of E-K aligned words were corrected than E-P aligned words as two example words shown in Fig. 6; many Korean "ㅡ" *PU*'s were correctly aligned like "b" - "ㅂㅡ" and "d" - "ㄷㅡ" alignment in the direct method.

We repeated the learning 30 times and we noticed that the character accuracy did not change much after 3 or 4 iterations. So we chose the parameters at 3rd iteration for the direct method and at 4th iteration for the pivot method.

Table 7. Iterative learning effect for *character accuracy* (decoding stack size=40)

iteration \ method	Direct			Pivot		
	seen	unseen	average	seen	unseen	average
1	70.6%	58.6%	64.6%	73.6%	59.0%	66.3%
2	82.2%	63.1%	72.7%	78.0%	60.5%	69.3%
3	82.0%	66.3%	**74.2%**	75.9%	61.1%	68.5%
4	79.8%	64.6%	72.2%	78.7%	61.9%	**70.3%**
5	80.5%	63.1%	71.8%	77.3%	60.3%	68.8%

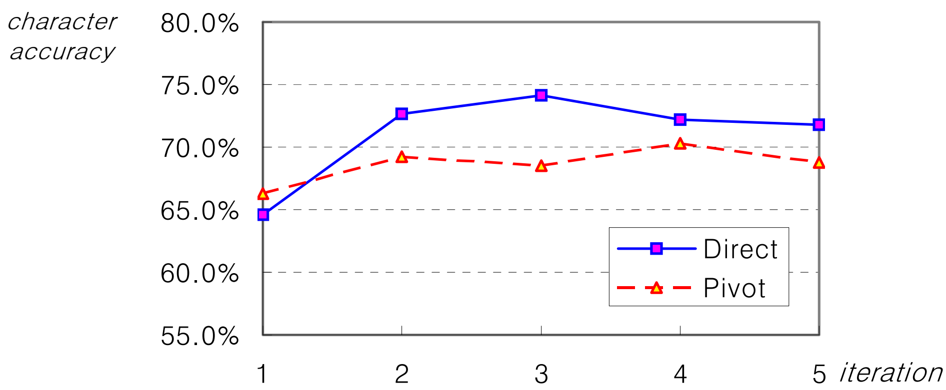

Fig. 5. Iterative learning effect (average *character accuracy*)

Direct method

broker

first alignment

b	r	o	k	er
ㅂ	ㅡㄹ	ㅗ	ㅋ	ㅓ

second alignment

b	ro	k	er
ㅂㅡ	ㄹㅗ	ㅋ	ㅓ

card

first alignment

ca	rd
ㅋㅏ	ㄷㅡ

second alignment

c	ar	d
ㅋ	ㅏ	ㄷㅡ

Pivot method

first alignment

b	ro	ke	r
b	ro	uk	ə

second alignment

br	o	k	er
br	ou	k	ə

first alignment

c	ar	d
k	ɑː	d

second alignment (no change)

c	ar	d
k	ɑː	d

Fig. 6. Alignment changes after 2 iterations

4.1.4 Test result

With the best parameters we found, we tested our model for the Test Set I, and the result is in Table 8. Both the *character accuracy* and *word accuracy* of the direct method were better than that of the pivot method by 3.9% and 4.7%.

Table 8. *Character accuracy* and *word accuracy* against Test Set I

acc. / method	*character accuracy*			*word accuracy* (for 20 generations)		
	seen	unseen	average	seen	unseen	average
Direct	82.0%	66.3%	74.2%	72.7%	40.7%	56.7%
Pivot	78.7%	61.9%	70.3%	64.7%	39.3%	52.0%

Direct: 2-*PU*, *dist*=0.4, 3 iterative learning
Pivot: 2-*PU*, *dist*=0.3, 4 iterative learning

4.1.5 Comparison with other systems

We did not use heuristics in order to keep our model language-independent. If we use some heuristics, the STM can be improved significantly. For example, we tested the model that uses the phonetic type to align the *PU's*: only same phonetic types (vowels, consonants and semi vowels) of *PU's* can be segmented and aligned together. It increased the performance of the direct method. (As the definition of phonetic type may be subjective and it did not increase the performance for the pivot method, we do not present this result to concentrate on the fair comparison of two methods.) The newly aligned data was manually but roughly aligned again and used to train our model. The model produced a better result: *character accuracy* was 78.5%, *word accuracy* for single generation was 38.1%, and *word accuracy* for 20 generations was 63.3%. The performance could have been improved upon if we had more carefully aligned the data. For the comparison with the rule-based system, we implemented the transliteration system of Kim [10] and tested it with our Test Set I: the *character accuracy* was 75.1% and *word accuracy* for single generation was 38.0%, which were lower than our result of the manually trained direct method.

Table 9. Average *character accuracy* against Test Set I

method	*Character accuracy*
Direct	74.2%
Pivot	70.3%
Manually modified Direct	78.5%
Kim	75.1%

Some other works for the transcription systems are found in NetTalk and MBRTalk [26]. They were trained with manually character-aligned pairs while we did it automatically from the word-aligned pairs. Lucas [17] used neural networks to learn the translation rule from the word-aligned pairs as we did with the STM, and 67% of his result was correct in symbol score[5]. For the comparison, we evaluated E-P conversion which was the first step in the pivot method. The average *character accuracy* was about 68.9% (seen=79.4%, unseen=58.4%), which was a little bit higher than that of neural networks approach. It shows that the STM is very efficient in learning rules from word-aligned pairs and can be used both for the transcription and the transliteration systems too.

4.2 Variation coverage

4.2.1 Test set

For the testing of real field usage, we constructed Test Set II that reflects the usage frequencies and diversity of transliterations. We extracted usage in KTSET 2.0 and made the variation list for 2,391 English words. The samples are already shown in Table 2. Notice that most of transliterations do not have many variations but have 1.14 various words in average. The variation list was constructed with the following procedures:

1. *Extract nouns and equivalents*: using the Korean morphological analyzer (MA) [8], we extracted all

[5] It is comparable to our character accuracy, but it is a little bit different from ours; *L* is the larger length of two strings in formula [17].

the nouns that could be index terms. Because most of the transliterations and their variations were not registered in MA's dictionary, the MA produced many of them as unknown words, which we considered as nouns.

2. *Select transliterations from the extracted nouns*: in order to identify transliterated nouns, we used a language model that was based on the formula (8) and (12). Two models were trained for the transliterations and the pure Korean words. The models calculated the probability of how much a word could be either a transliteration or a pure Korean word. If the word had the higher probability for transliteration word language model than for pure Korean word language model, it was selected as a transliteration[6]. This process extracted 5,394 transliteration token types, which was 17.4% of all the nouns. We already showed the statistics in Table 1.
3. *Extract pure transliteration parts*: some Korean words and suffixes were contained in the selected words. We extracted manually the pure transliteration parts from them.
4. *Find an original English word of the transliterations*: we manually found the original word of the transliterations. During this process, we made two decisions: 1. some words like special names, which were hard to find their original words, were dropped; 2. original words of some transliterations, which were ambiguous, were chosen arbitrarily. For example, the original word of "파일 (phail)" may be either "file" or "pile" depending on the context. We chose arbitrarily "file" in this case. More delicate process should look up the context to determine the correct one.
5. *List the transliterations for the same original words*: once the original English words were found, the pairs were sorted in the English word order and merged together for the same original words. The result was the list of various transliterations.

4.2.2 Evaluation function

For real usage testing, we used *variation coverage*, which considers various usage and counts the usage frequency in the evaluation. If a transliteration is used more frequently than the others, it is counted more. We evaluated both for the term frequency (*tf*) and document frequency (*df*), where *tf* is the number of term appearance in the documents and *df* is the number of documents that contain the term.

$$variation\ coverage = \frac{total\ frequency\ of\ found\ words}{total\ frequency\ of\ usage\ words} \tag{18}$$

4.2.3 Test result of the two methods

For the 2,391 English words extracted during the Test Set II construction, their transliterations were generated in probability order. The transliterations were compared one by one with the

[6] For the training, the data in Test Set I was used for the transliteration language model, and the pure Korean words extracted from a Korean dictionary were used for the pure Korean word language model.

transliteration usage. If they were matched, their usage *tf* (or their usage *df*) was added to the total frequency of the found words. If we set the usage *tf* (or *df*) of the transliterations to 1 for each transliteration, we can calculate the transliteration coverage for the unique word types: we name it as *single frequency* coverage, in short *sf*. The variation coverage for *tf*, *df* and *sf* for both methods are shown in Table 10.

We can see that the direct method was superior to the pivot method in all criteria. The performance gap was getting larger as we considered frequencies less: gap for *tf* was 2.7%, *df* was 4.9%, and s*f* was 6.2%. It means that the direct method generates more various transliterations than the pivot method; in other words, the pivot method generates relatively more frequently used transliterations.

Table 10. *Variation coverage* against Test Set II (Direct and Pivot method)

	tf	*df*	*sf*
Direct	52.4%	53.4%	37.0%
Pivot	49.7%	48.5%	30.8%

Table 11 shows the transliteration usage and their *tf*. The transliteration types of the usage are identified with the postfix "[W]" for the written word transliteration and "[S]" for the spoken word transliteration respectively. Table 12 and Table 13 show the *tf variation coverage* of the transliterations that were generated by the two methods for the English words in Table 11.

Table 11. English words and their transliteration usage found in KTSET 2.0

English	usage 1	*tf*	usage 2	*tf*	usage 3	*tf*	total *tf*
data	데이타 (teyitha) [W]	933	데이터 (teyithe) [S]	563			1496
digital	디지틀 (ticithul) [S]	269	디지탈 (ticithal) [W]	246	디지털 (ticithel) [S]	4	519
system	시스템 (sisutheym) [W]	3398	시스팀 (sisuthim) [S]	2487	시스탬 (sisuthaym) [W]	1	5886
service	서비스 (sepisu) [S]	1658					1658
mode	모드 (motu) [S]	117					117
asia	아시아 (asia) [W]	125	아세아 (aseya) [W]	6			131
radio	라디오 (latio) [W]	26	레이디오 (leyitio) [S]	3			29

The words with postfix [W] are transliterated from written words and those with [S] are from spoken words respectively.

We roughly tested this method; it selected 98% of transliterations from the training data.

Table 12. Top 5 transliterations generated by direct method STM

English	1st	2nd	3rd	4th	5th	total *tf*	coverage
data	**데이터 (teyithe) [S]**	달타 (taltha)	더타 (tetha)	데타 (teytha)	**데이타 (teyitha) [W]**	1496	100%
digital	디지터 (ticithe)	**디지털 (ticithel) [S]**	**디지탈 (ticithal) [W]**	디지아 (ticia)	디지타 (ticitha)	250	48%
system	**시스템 (sisuthem) [W]**	시스텀 (sisutheym)	사이스템 (saisutheym)	지스템 (cisuthem)	지스텀 (cisuthem)	3398	58%
service	사이비스 (saipisu)	서비 (sepi)	사이비서 (saipise)	시비 (sipi)	지비스 (cipisu)	0	0%
mode	모데 (motey)	모덱 (moteyk)	마드 (matu)	마디 (mati)	어드 (etu)	0	0%
asia	**아시아 (asia) [W]**	에시아 (eysia)	애시아 (aysia)	아자 (aca)	에이시아 (eyisia)	125	95%
radio	**라디오 (latio) [W]**	로디오 (lotio)	래비오 (laypio)	로다이어 (lotaie)	레이더 (leyite)	26	90%

The words with postfix [W] are transliterated from written words and those with [S] are from spoken words respectively.

Table 13. Top 5 transliterations generated by pivot method STM

English	1st	2nd	3rd	4th	5th	total *tf*	coverage
data	**데이터 (teyithe) [S]**	데이트 (teyithu)	데이트르 (teyithulu)	다티 (tathi)	다터 (tathe)	563	38%
digital	디자이틀 (ticaithul)	디자이토 (ticaitho)	**디지틀 (ticithul) [S]**	디지토 (ticitho)	디자이털 (thicaithel)	269	52%
system	시셈 (siseym)	**시스팀 (sisuthim) [S]**	시스터 (sisuthe)	시심 (sisim)	시스티즈 (sisuthicu)	2487	42%
service	**서비스 (sepisu) [S]**	시비스 (sepisu)	서빈스 (sepinsu)	서빅 (sepik)	서비즈 (sepicu)	1658	100%
mode	마더 (mate)	마드 (matu)	모더 (mote)	**모드 (motu) [S]**	마지 (maci)	117	100%
asia	애시어 (aysie)	에이시어 (eyisie)	애지어 (aycie)	에이사이 (eyisai)	어시어 (esie)	0	0%
radio	레디오 (leytio)	레이디어 (leyitie)	러진 (lecin)	러드 (letu)	레이다이 (leyitai)	0	0%

The words with postfix [W] are transliterated from written words and those with [S] are from spoken words respectively.

4.2.4 Hybrid method for variation coverage

The direct method was trained with words of the standard usage which contain both written word transliteration and spoken word transliteration. Therefore, it should produce both transliterations. We see an example in Table 12 and 13; For the word “data”, the direct method produced “데이터 (teyithe) [S]” (spoken word transliteration) at the first generation and “데이타 (teyitha) [W]” (written word transliteration) at the fifth generation. In comparison, the pivot method produced only “데이터 (teyithe) [S]” at the first generation. However, it does not mean that the direct method always finds all kinds of transliterations; the direct method transliterated “digital” to “디지털 (ticithel) [S]” and “디지탈 (ticithal) [W]”, but the “디지틀 (ticithul) [S]” was generated by the pivot method. Generally, we can conjecture that many written word transliterations are found by the direct method, and many spoken word transliterations are found by the pivot method. For examples, in Table 12 and Table 13, the “서비스 (sepisu) [S]” and “모드 (motu) [S]” were generated by the pivot method, but not by the direct method, while the “아시아 (asia) [W]” and “라디오 (latio) [W]” were found by the direct method, but not by the pivot method.

In order to exploit the benefits of the two methods, we used a hybrid method; the pivot method is used to find spoken transliteration usage and the direct method is for written transliteration usage. Merging the two outputs and discarding duplicates become a new output of the hybrid method. For the evaluation, we took top 10 generations from each method to make 20 or less generations. The result was much better than the two methods as shown in Table 14 and in Fig. 7. It means that two types of transliterations are used in real field documents and each model produces the different transliterations.

Table 14. *Variation coverage* against Test Set II (Hybrid, Direct and Pivot method)

	tf	*df*	*sf*
Hybrid	76.0%	73.9%	47.1%
Direct	52.4%	53.4%	37.0%
Pivot	49.7%	48.5%	30.8%

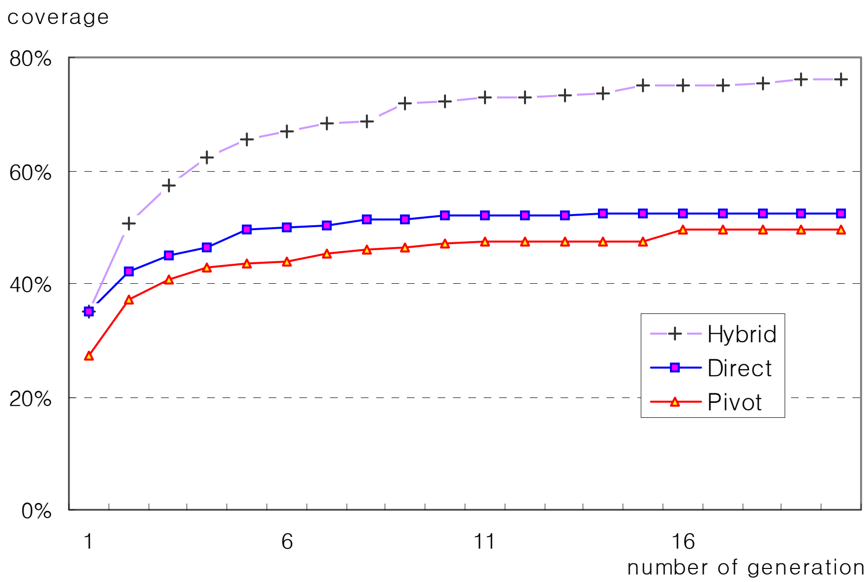

Fig. 7. Term frequency coverage of various transliterations

4.3 Information retrieval performance

4.3.1 Query expansion

Because we did not have a test set for English-to-Korean cross lingual information retrieval at the time we tested our methods, we used KTSET 2.0 with modified queries for our testing; we reverse-transliterated the transliterations to the original English words in the 32 queries out of the total 50 queries. The others were used as they were in the original set.

After tokenizing the modified natural language queries, we expanded the queries using the three methods. We generated 4 transliterations per one English word and did not remove the duplicated transliterations that might be generated during transliteration. Table 15 shows the example of a query and its expanded queries. None of the transliterations for "multimedia" was matched with the transliteration "멀티미디어 (melthimitie)" in the original query. However, many of them were close to the pronunciation of the transliteration. By contrast, the transliteration for the "printer" contained the same transliteration word in the original query.

Table 15. An example of query expansion

Original query	멀티미디어를 위한 VGA 나 프린터에 대해 알고 싶습니다. (I would like to know about VGA or a printer for multimedia.)
Reverse transliterated query	multimedia 를 위한 VGA 나 printer 에 대해 알고 싶습니다.
Tokenized query	multimedia, VGA, printer
Expanded query using Direct method	멀타이메디어(melthaimeytie), 멀타이미디어(melthaimitie), 멀타이메이아 (melthaimeyia) , 멀타이미디아(melthaimitia), multimedia, VGA, 프린터(phulinthe), 프린터(phulinthe), 프린트(phulinthu), 프린처(pulinche), printer
Expanded query using Pivot method	몰티미디에이(molthimitieyi), 몰티메디에이(molthimeythieyi), 몰티메이어 (molthimeyie), 몰티메이디에이(molthimeyitieyi), multimedia, VGA, 프린터 (phulinthe), 프린터르(phulinthelu), 프린터(phulinthe), 프린트르(phulinthulu), printer
Expanded query using Hybrid method	멀타이메디어(melthaimeytie), 몰티미디에이(molthimitieyi), 멀타이미디어 (melthaimitie), 몰티메디에이(molthimeythieyi), multimedia, VGA, 프린터 (phulinthe), 프린터(phulinthe), 프린터(phulinthe), 프린터르(phulinthelu), printer

4.3.2 Evaluation

Using the SMART system, we evaluated the effectiveness of the query expansion methods. SMART uses the vector space model and retrieves the documents based on term vector similarity [22]. Each term in documents and queries are assumed to be independent of each other and the similarity between a document and a query is calculated as the inner product between weighted term vectors. The term weight is usually represented as term frequency. When a document, *d*, is represented in a weighted term vector (w_{d1}, w_{d2}, ... w_{dn}) and a query, *q*, (w_{q1}, w_{q2}, ... w_{qn}), the similarity is calculated as follows:

$$Sim(d,q) = \sum_{i=1}^{n} (w_{di} \times w_{qi}) \qquad (19)$$

In our experiment, we used the following weighting formula that was used by Salton [23].

$$w_{ik} = \frac{(0.5 + 0.5\frac{tf_{ik}}{\max tf}) \bullet \ln \frac{N}{df_k}}{\sum_{k=1}^{n}((0.5 + 0.5\frac{tf_{ik}}{\max tf}) \bullet \ln \frac{N}{df_k})^2} \tag{20}$$

where tf_{ik} is the occurrence frequency of term t_k in document d_i, N is the collection size, and df_k is the number of documents in which term t_k appears.

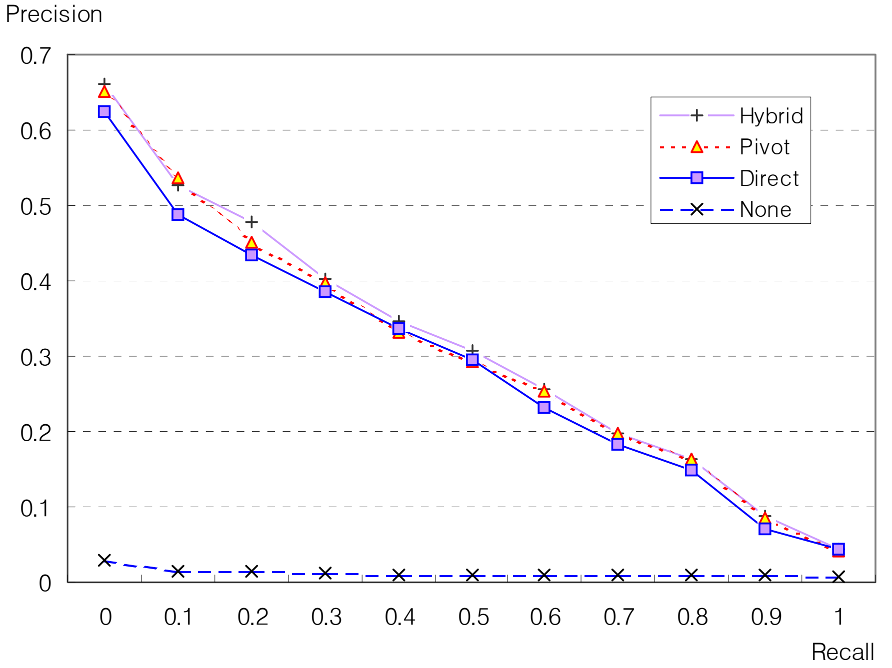

Fig 8. Recall-precision graph for KTSET 2.0

Table 16. 11-point average of recall-precision

methods	11-pt average
Hybrid	0.3156
Pivot	0.3092
Direct	0.2945
None	0.0126

The relevance graph is shown in Figure 8. The "None" means the result of the tokenized query. It naturally shows that using the English token is not effective for Korean documentation retrieval. Table 16 shows the 11-point average. Being different from the previous experiment results, this experiment showed that the pivot method performed better than the direct method. We conjecture that because the pivot method usually produces more frequently used transliterations and the transliterations in queries of KT SET 2.0 are more frequently used ones, the pivot method was more adequate in finding the most relevant documents than the direct method. Considering that the KT SET 2.0 contains rather formal documents like technical papers, technical reports and articles, more various transliterations may be found in an open environment as in informal email communications and internet web sites. Consequently, we may conjecture that the direct method may be more effective in the open environment. Regardless of this result, in this experiment, the hybrid method performed the best; 7.16% performance increased over the direct method, and 2.07% over the pivot method. We can conjecture that the hybrid method takes best transliterations from each method to find the relevant documents effectively.

5. Conclusion

We have presented two methods for the E-K transliteration based on the statistical model. The direct method slightly performed better than the pivot method both in the *accuracy* and the *variation coverage*. However, the pivot method performed better than the direct method in the information retrieval effectiveness test. Nevertheless, both methods have a limitation in the performance because transliterations in Korean documents are not used in a consistent way. Our experiment showed that the hybrid method, which is a combination of the pivot and the direct method, was more efficient to find the most probable transliterations and the most relevant documents than each of the single methods.

The model that we used here to compare two methods was very simple because we wanted to keep it language independent as much as possible. A slightly modified model improved its performance and worked better than other systems: rule-based E-K transliteration [10] and neural networks transcription [17]. More delicate E-K transliteration systems for the pivot and the direct method can be developed based on the STM. Its hybrid version can be used effectively to find transliteration variations for information retrieval system.

References

[1] L. R. Bahl, F. Jelinek and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, March, 1983.

[2] P. F. Brown and *et al*, "A Statistical Approach to Machine Translation," *Computational Linguistics*, Vol. 16, No. 2, June, 1990.

[3] P. F. Brown and *et al*, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, Vol. 19, No. 2, 1993.

[4] N. Collier, A. Kumano and H. Hirakawa, "Acquisition of English-Japanese proper nouns from

noisy-parallel newswire articles using Katakana matching," *Natural Language Processing Pacific Rim Symposium '97*, 1997.

[5] D. A. Hull and G. Grefenstette, "Querying Across Languages: A Dictionary-Based Approach to Multilingual Information Retrieval," *ACM SIGIR Conference on Information Retrieval*, 1996.

[6] K. Jeong, Y. Kwon and S. H. Myaeng, "The Effect of a Proper Handling of Foreign and English Words in Retrieving Korean Text," in *Proceedings of the 2nd International Workshop on Information Retrieval with Asian Languages - 1997*, Tsukuba-shi, Japan, 1997.

[7] K. Jeong, Y. Kwon and S. H. Myaeng, "Construction of Equivalence Classes of Foreign Words through Automatic Identification and Extraction," *Natural Language Processing Pacific Rim Symposium '97*, 1997.

[8] S. S. Kang, "HAM: Korean Morphological Analyzer and Automatic Indexing System," http://ham.hansung.ac.kr/ham-intr.html, 1996. (in Korean)

[9] Y. Kang and A. A. Maciejewski, "An Algorithm for Generating a Dictionary of Japanese Scientific Terms," *Literary and Linguistic Computing*, Vol. 11, No. 2, 1996.

[10] B. H. Kim, "Automatic Transliteration of the English Words into Hangul," Master dissertation, Public Policy Graduated School, Seogang University, 1991. (in Korean)

[11] J. Kim, Y. Kim and S. Kim, "Development of the Data Collection (KTSET) for Korean Information Retrieval Studies," *The 6th Hangul and Korean Information Processing*, 1994.

[12] Y. H. Kwon, K. S. Jeong and S. H. Myaeng, "Foreign Word Identification Using a Statistical Method for Information Retrieval," *Proceedings of the Seventeenth International Conference on Computer Processing of Oriental Languages*, Hong Kong, 1997.

[13] H. B. Lee, "On Translitering Loan Words by the Korean Alphabet - A Critical Appraisal of the Revised Koreanization System -," *Linguistics Research*, Vol. 15, No. 1, 1979. (in Korean)

[14] H. S. Lee and B. H. An, *Hangul Spelling Rule Lecture - updated edition*, Shinku Moonwhasa, 1994. (in Korean)

[15] J. S. Lee and K. S. Choi, "Two approaches to the Roman to Hangul Transcription Based on Statistical Translation Method," *Proceedings of the Seventeenth International Conference on Computer Processing of Oriental Languages*, Hong Kong, 1997.

[16] J. S. Lee and K. S. Choi, "A Statistical Method to Generate Various Foreign Word Transliterations in Multilingual Information Retrieval System," *Proceedings of the 2nd International Workshop on Information Retrieval with Asian Languages - 1997*, Tsukuba-shi, Japan, 1997.

[17] S. M. Lucas and R. I. Damper, "Syntactic neural networks for bi-directional text-phonetics," pp. 127-141 in *Talking Machines*, ed. G. Bailly and C. Benoit, North Holland, 1991.

[18] M. Mettler, "TRW Japanese Fast Data Finder," *TIPSTER Text Program Phase I Proc.*, 1993, pp. 113-116.

[19] S. H. Parfitt and R. A. Sharman, "A Bi-directional Model of English Pronunciation," *EuroSpeech*, 1991.

[20] Y. C. Park, K. Choi, J. Kim, and Y. Kim, "Development of the Data Collection Ver. 2.0 (KTSET 2.0) for Korean Information Retrieval Studies," *Artificial Intelligence Spring Conference*, Korea Information Science Society, 1996. (in Korean)

[21] L. Rabiner, "Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, Vol. 77, No. 2, 1989, pp. 257-286.

[22] G. Salton, A. Wong and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of ACM*, Vol. 18, No. 11, 1975, pp. 613-620.

[23] G. Salton and C. Buckley, "Term-Weighting Approaches In Automatic Text Retrieval," *Information Processing & Management*, Vol. 24, No. 5, 1988, pp. 513-523.

[24] SERI/KIST, "A Study on the Development of Intelligent Information Processor," *1st year final report, Minister of science and technology*, 1995, pp. 366-383. (in Korean)

[25] P. Sheridan and J. P. Ballerini, "Experiments in Multilingual Information Retrieval using the SPIDER system," *ACM SIGIR Conference on Information Retrieval*, 1996.

[26] C. Stanfill and D. Waltz, "Toward Memory-Based Reasoning," *Communications of the ACM*, Vol. 29, No. 12, 1986.