English-to-Korean Transliteration using Multiple Unbounded Overlapping Phoneme Chunks

In-Ho Kang and **GilChang Kim**
Department of Computer Science
Korea Advanced Institute of Science and Technology

Abstract

We present in this paper the method of English-to-Korean(E-K) transliteration and back-transliteration. In Korean technical documents, many English words are transliterated into Korean words in various forms in diverse ways. As English words and Korean transliterations are usually technical terms and proper nouns, it is hard to find a transliteration and its variations in a dictionary. Therefore an automatic transliteration system is needed to find the transliterations of English words without manual intervention.

To explain E-K transliteration phenomena, we use phoneme chunks that do not have a length limit. By applying phoneme chunks, we combine different length information with easy. The E-K transliteration method has three steps. In the first, we make a phoneme network that shows all possible transliterations of the given word. In the second step, we apply phoneme chunks, extracted from training data, to calculate the reliability of each possible transliteration. Then we obtain probable transliterations of the given English word.

1 Introduction

In Korean technical documents, many English words are used in their original forms. But sometimes they are transliterated into Korean in different forms. Ex. 1, 2 show the examples of various transliterations in KTSET 2.0(Park et al., 1996).

(1) data

(a) 데이타(teyitha) [1,033][1]

(b) 데이터(teyithe) [527]

[1]the frequency in KTSET

(2) digital

(a) 디지틀(ticithul) [254]

(b) 디지탈(tichithal) [7]

(c) 디지털(ticithel) [6]

These various transliterations are not negligible for natural language processing, especially in information retrieval. Because same words are treated as different ones, the calculation based on the frequency of word would produce misleading results. An experiment shows that the effectiveness of information retrieval increases when various forms including English words are treated equivalently(Jeong et al., 1997).

We may use a dictionary, to find a correct transliteration and its variations. But it is not feasible because transliterated words are usually technical terms and proper nouns that have rich productivity. Therefore an automatic transliteration system is needed to find transliterations without manual intervention.

There have been some studies on E-K transliteration. They tried to explain transliteration as phoneme-per-phoneme or alphabet-per-phoneme classification problem. They restricted the information length to two or three units before and behind an input unit. In fact, many linguistic phenomena involved in the E-K transliteration are expressed in terms of units that exceed a phoneme and an alphabet. For example, '*a*' in '*ace*' is transliterated into "에 이*(eyi)*" but in '*acetic*', "어*(e)*" and in '*acetone*', "아*(a)*". If we restrict the information length to two alphabets, then we cannot explain these phenomena. Three words get the same result for '*a*'.

(3) ace 에이스(eyisu)

(4) acetic 어시틱(esithik)

(5) acetone 아세톤(aseython)

In this paper, we propose the E-K transliteration model based on phoneme chunks that do not have a length limit and can explain transliteration phenomena in some degree of reliability. Not a alphabet-per-alphabet but a chunk-per-chunk classification problem.

This paper is organized as follows. In section 2, we survey an E-K transliteration. In section 3, we propose phoneme chunks based transliteration and back-transliteration. In Section 4, the results of experiments are presented. Finally, the conclusion follows in section 5.

2 English-to-Korean transliteration

E-K transliteration models are classified in two methods: the pivot method and the direct method. In the pivot method, transliteration is done in two steps: converting English words into pronunciation symbols 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 without intermediate steps. An experiment shows that the direct method is better than the pivot method in finding variations of a transliteration(Lee and Choi, 1998). Statistical information, neural network and decision tree were used to implement the direct method.

2.1 Statistical Transliteration method

An English word is divided into phoneme sequence or alphabet sequence as $e_1, e_2, \ldots, e_n$. Then a corresponding Korean word is represented as $k_1, k_2, \ldots, k_n$. If a corresponding Korean character (k_i) does not exist, we fill the blank with '-'. For example, an English word "dressing" and a Korean word "드레싱(tuleysing)" are represented as Fig. 1. The upper one in Fig. 1 is divided into an English phoneme unit and the lower one is divided into an alphabet unit.

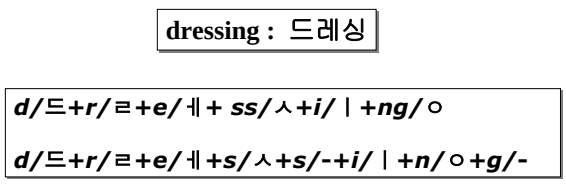

Figure 1: An E-K transliteration example

The problem in statistical transliteration method is to find out the most probable transliteration 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 $p(K|E)$. By using the Bayes' theorem, we can rewrite the transliteration problem as follows:

$$arg \max_K p(K|E) = arg \max_K p(K)p(E|K) \quad (1)$$

With the Markov Independent Assumption, we approximate $p(K)$ and $p(E|K)$ as follows:

$$p(K) \cong p(k_1)\prod_{i=2}^{n} p(k_i|k_{i-1}) \quad (2)$$

$$p(E|K) \cong \prod_{i=1}^{n} p(e_i|k_i) \quad (3)$$

As we do not know the pronunciation of a given word, we consider all possible phoneme sequences. For example, 'data' has following possible phoneme sequences, 'd-a-t-a, d-at-a, da-ta, ...'.

As the history length is lengthened, we can get more discrimination. But long history information causes a data sparseness problem. In order to solve a sparseness problem, Maximum Entropy Model, Back-off, and Linear interpolation methods are used. They combine different statistical estimators. (Tae-il Kim, 2000) use up to five phonemes in feature function(Berger et al., 1996). Nine feature functions are combined with Maximum Entropy Method.

2.2 Neural Network and Decision Tree

Methods based on neural network and decision tree deterministically decide a Korean character for a given English input. These methods take two or three alphabets or phonemes as an input and generate a Korean alphabet or phoneme as an output. (Jung-Jae Kim, 1999) proposed a neural network method that uses two surrounding phonemes as an input. (Kang, 1999) proposed a decision tree method that uses six surrounding alphabets. If an input does not cover the phenomena of proper transliterations, we cannot get a correct answer.

Even though we use combining methods to solve the data sparseness problem, the increase of an information length would double the complexity and the time cost of a problem. It is not easy to increase the information length. To avoid these difficulties, previous studies does not use previous outputs(k_{i-1}). But it loses good information of target language.

Our proposed method is based on the direct method to extract the transliteration and its variations. Unlike other methods that determine a certain input unit's output with history information, we increase the reliability of a certain transliteration, with known E-K transliteration phenomena (phoneme chunks).

3 Transliteration using Multiple unbounded overlapping phoneme chunks

For unknown data, we can estimate a Korean transliteration from hand-written rules. We can also predict a Korean transliteration with experimental information. With known English and Korean transliteration pairs, we can assume possible transliterations without linguistic knowledge. For example, '*scalar*' has common part with '*scale:*스케일*(sukheyil)*', '*casino:*카지노*(khacino)*', '*koala:*코알라*(khoalla)*', and '*car:*카*(kha)*' (Fig. 2). We can assume possible transliteration with these words and their transliterations. From '*scale*' and its transliteration 스케일(sukheyil), the '*sc*' in '*scalar*' can be transliterated as '스ㅋ(sukh)'. From a '*casino*' example, the '*c*' has more evidence that can be transliterated as 'ㅋ(kh)'. We assume that we can get a correct Korean transliteration, if we get useful experimental information and their proper weight that represents reliability.

3.1 The alignment of an English word with a Korean word

We can align an English word with its transliteration in alphabet unit or in phoneme unit. Korean vowels are usually aligned with English vowels and Korean consonants are aligned with English consonants. For example, a Korean consonant, 'ㅂ(p)' can be aligned with English consonants 'b', 'p', and 'v'. With this heuristic we can align an English word with its transliteration in an alphabet unit and a phoneme unit with the accuracy of 99.4%(Kang, 1999).

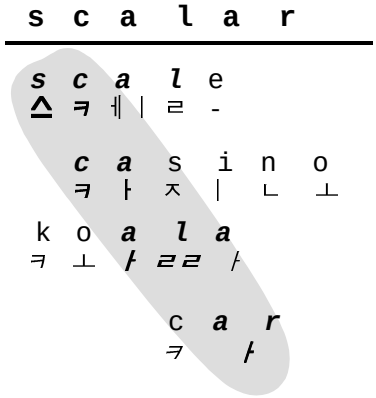

Figure 2: the transliteration of 'scalar : 스칼라(sukhalla)'

3.2 Extraction of Phoneme Chunks

From aligned training data, we extract phoneme chunks. We enumerate all possible subsets of the given English-Korean aligned pair. During enumerating subsets, we add start and end position information. From an aligned data "*dressing*" and "드레싱*(tuleysing)*", we can get subsets as Table 1[2].

Table 1: The extraction of phoneme chunks

Context	Output
@d	*d/*드*(d)*
d	*d/*드*(d)*
@dr	*d/*드*(d)+r/*르*(r)*
r	*r/*르*(r)*
@dre	*d/*드*(d)+r/*르*(r)+e/*ㅖ *(ey)*

The *context* stands for a given English alphabets, and the *output* stands for its transliteration. We assign a proper weight to each phoneme chunk with Equation 4.

$$weight(context : output) = \frac{C(output)}{C(context)} \quad (4)$$

$C(x)$ means the frequency of x in training data. Equation 4 shows that the ambiguous phenomenon gets the less evidence. The chunk weight is transmitted to each phoneme symbol. To compensate for the length of phoneme, we multiply the length of phoneme to the weight of the phoneme chunk(Fig. 3).

[2]@ means the start and end position of a word

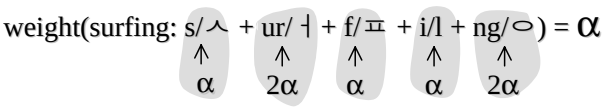

Figure 3: The weight of a chunk and a phoneme

This chunk weight does not mean the reliability of a given transliteration phenomenon. We know real reliability, after all overlapping phoneme chunks are applied. The chunk that has some common part with other chunks gives a context information to them. Therefore a chunk is not only an input unit but also a means to calculate the reliability of other chunks.

We also extract the connection information. From aligned training data, we obtain all possible combinations of Korean characters and English characters. With this connection information, we exclude impossible connections of Korean characters and English phoneme sequences. We can get the following connection information from "dressing" example(Table 2).

Table 2: Connection Information

English		*Korean*	
left	*right*	*left*	*right*
@	*d*	@	드*(d)*
d	*r*	드*(d)*	르*(r)*
r	*e*	르*(r)*	ㅖ*(ey)*

3.3 A Transliteration Network

For a given word, we get all possible phonemes and make a Korean transliteration network. Each node in a network has an English phoneme and a corresponding Korean character. Nodes are connected with sequence order. For example, '*scalar*' has the Korean transliteration network as Fig. 4. In this network, we disconnect some nodes with extracted connection information.

After drawing the Korean transliteration network, we apply all possible phoneme chunks to the network. Each node increases its own weight with the weight of phoneme symbol in a phoneme chunks (Fig. 5). By overlapping the weight, nodes in the longer chunks get more evidence. Then we get the best path that has the highest sum of weights, with the Viterbi algorithm. The Tree-Trellis algorithm is used to get the variations(Soong and Huang, 1991).

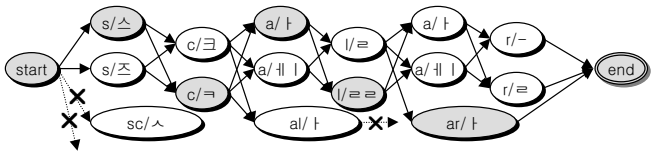

Figure 4: Korean Transliteration Network for 'scalar'

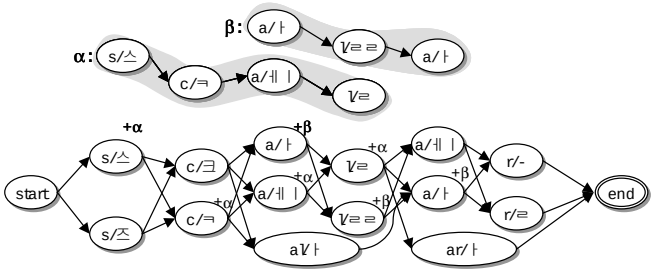

Figure 5: Weight application example

4 E-K back-transliteration

E-K back transliteration is a more difficult problem than E-K transliteration. During the E-K transliteration, different alphabets are treated equivalently. For example, 'f, p' and 'v, b' are transliterated into 'ㅍ(ph)' and 'ㅂ(p)' respectively and the long sound and the short sound are also treated equivalently. Therefore the number of possible English phonemes per a Korean character is bigger than the number of Korean characters per an English phoneme. The ambiguity is increased. In E-K back-transliteration, Korean phonemes and English phonemes switch their roles. Just switching the position. A Korean word is aligned with an English word in a phoneme unit or a character unit(Fig. 6).

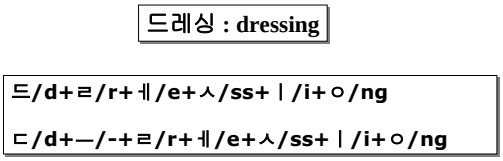

Figure 6: E-K back-transliteration example

5 Experiments

Experiments were done in two points of view: the accuracy test and the variation coverage test.

5.1 Test Sets

We use two data sets for an accuracy test. *Test Set I* is consists of 1,650 English and Korean word pairs that aligned in a phoneme unit. It was made by (Lee and Choi, 1998) and tested by many methods. To compare our method with other methods, we use this data set. We use same training data (1,500 words) and test data (150 words). *Test Set II* is consists of 7,185 English and Korean word pairs. We use *Test Set II* to show the relation between the size of training data and the accuracy. We use 90% of total size as training data and 10% as test data. For a variation coverage test, we use *Test Set III* that is extracted from KTSET 2.0. *Test Set III* is consists of 2,391 English words and their transliterations. An English word has 1.14 various transliterations in average.

5.2 Evaluation functions

Accuracy was measured by the percentage of the number of correct transliterations divided by the number of generated transliterations. We call it as *word accuracy*(*W.A.*). We use one more measure, called *character accuracy*(*C.A.*) that measures the character edit distance between a correct word and a generated word.

$$W.A. = \frac{no.\ of\ correct\ words}{no.\ of\ generated\ words} \quad (5)$$

$$C.A. = \frac{L-(i+d+s)}{L} \quad (6)$$

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(Hall and Dowling, 1980).

For the real usage test, we used variation coverage ($V.C.$) that considers various usages. 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. 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, *single frequency*(*sf*).

$$V.C. = \frac{\{tf, df, sf\}\ of\ found\ words}{\{tf, df, sf\}\ of\ used\ words} \quad (7)$$

5.3 Accuracy tests

We compare our result [*PCa, PCp*][3] with the simple statistical information based model(Lee and Choi, 1998) [*ST*], the Maximum Entropy based model(Tae-il Kim, 2000) [*MEM*], the Neural Network model(Jung-Jae Kim, 1999) [*NN*] and the Decision Tree based model(Kang, 1999)[*DT*]. Table 3 shows the result of E-K transliteration and back-transliteration test with *Test Set I*.

Table 3: *C.A.* and *W.A.* with *Test Set I*

	E-K trans.		E-K back trans.	
method	*C.A.*	*W.A.*	*C.A.*	*W.A.*
ST	69.3%	40.7%[4]	60.5%	-
MEM	72.3%	43.3%	-	-
NN	79.0%	35.1%	-	-
DT	78.1%	37.6%	77.1%	31.0%
PCp	86.5%	55.3%	81.4%	34.7%
PCa	85.3%	46.7%	79.3%	32.6%

Fig. 7, 8 show the results of our proposed method with the size of training data, *Test Set II*. We compare our result with the decision tree based method.

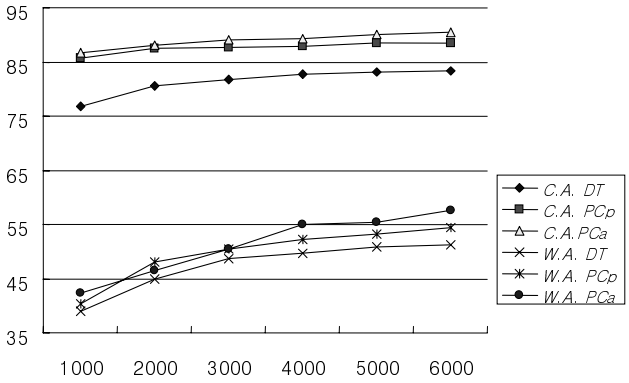

Figure 7: E-K transliteration results with *Test Set II*

[3] *PC* stands for phoneme chunks based method and *a* and *b* stands for aligned by an alphabet unit and a phoneme unit respectively

[4] with 20 higher rank results

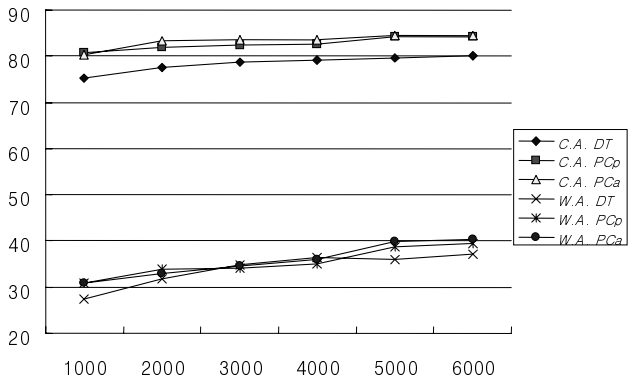

Figure 8: E-K back-transliteration results with *Test Set II*

With *Test Set II*, we can get the following result (Table 4).

Table 4: *C.A.* and *W.A.* with the *Test Set II*

	E-K trans.		E-K back trans.	
method	*C.A.*	*W.A.*	*C.A.*	*W.A.*
PCp	89.5%	57.2%	84.9%	40.9%
PCa	90.6%	58.3%	84.8%	40.8%

5.4 Variation coverage tests

To compare our result(PCp) with (Lee and Choi, 1998), we trained our methods with the training data of *Test Set I*. In ST, (Lee and Choi, 1998) use 20 high rank results, but we just use 5 results. Table 5 shows the coverage of our proposed method.

Table 5: variation coverage with *Test Set III*

method	*tf*	*df*	*sf*
ST	76.0%	73.9%	47.1%
PCp	84.0%	84.0%	64.0%

Fig. 9 shows the increase of coverage with the number of outputs.

5.5 Discussion

We summarize the information length and the kind of information(Table 6). The results of experiments and information usage show that *MEM* combines various information better than *DT* and *NN*. *ST* does not use a previous input (e_{i-1}) but use a previous output(k_{i-1}) to calculate the current output's probability like

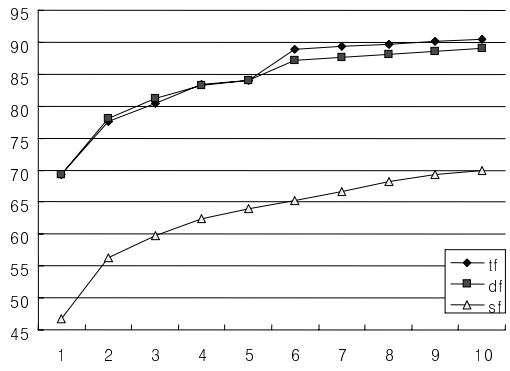

Figure 9: The *V.C.* result

Table 6: Information Usage

method	*before*	*behind*	*previous output*
ST	2	0	Y
MEM	2	2	N
NN	1	1	N
DT	3	3	N
PC	-	-	Y

Part-of-Speech Tagging problem. But *ST* gets the lowest accuracy. It means that surrounding alphabets give more information than previous output. In other words, E-K transliteration is not the alphabet-per-alphabet or phoneme-per-phoneme classification problem. A previous output does not give enough information for current unit's disambiguation. An input unit and an output unit should be extended. E-K transliteration is a chunk-per-chunk classification problem.

We restrict the length of information, to see the influence of phoneme-chunk size. Fig. 10 shows the results.

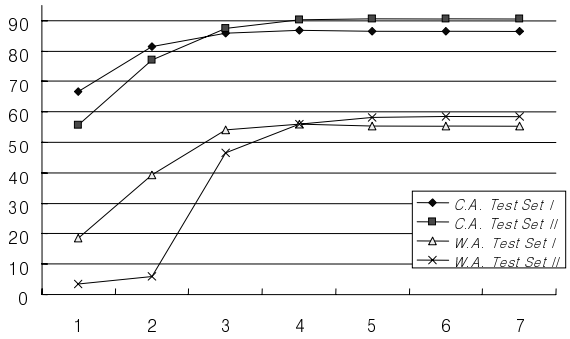

Figure 10: the result of a length limit test

With the same length of information, we get the higher C.A. and W.A. than other methods. It means previous outputs give good information and our chunk-based method is a good combining method. It also suggests that we can restrict the max size of chunk in a permissible size.

PCa gets a higher accuracy than PCp. It is due to the number of possible phoneme sequences. A transliteration network that consists of phoneme unit has more nodes than a transliteration network that consists of alphabet unit. With small training data, despite of the loss due to the phoneme sequences ambiguity a phoneme gives more information than an alphabet. When the information is enough, PCa outperforms PCp.

6 Conclusions

We propose the method of English-to-Korean transliteration and back-transliteration with multiple unbounded overlapping phoneme chunks. We showed that E-K transliteration and back-transliteration are not a phoneme-per-phoneme and alphabet-per-alphabet classification problem. So we use phoneme chunks that do not have a length limit and can explain E-K transliteration phenomena. We get the reliability of a given transliteration phenomenon by applying overlapping phoneme chunks. Our method is simple and does not need a complex combining method for various length of information. The change of an information length does not affect the internal representation of the problem. Our chunk-based method can be used to other classification problems and can give a simple combining method.

References

Tae-il Kim. 2000. English to Korean transliteration model using maximum entropy model for cross language information retrieval. Master's thesis, Seogang University (*in Korean*).

Kil Soon Jeong, Sung Hyun Myaeng, Jae Sung Lee, and Key-Sun Choi. 1999. Automatic identification and back-transliteration of foreign words for information retrieval. *Information Processing and Management.*

Key-Sun Choi Jung-Jae Kim, Jae Sung Lee. 1999. Pronunciation unit based automatic English-Korean transliteration model using neural network. In *Proceedings of Korea Cognitive Science Association(in Korean).*

Byung-Ju Kang. 1999. Automatic Korean-English back-transliteration. In *Proceedings of the 11th Conference on Hangul and Korean Language Information Processing(in Korean).*

Jae Sung Lee and Key-Sun Choi. 1998. English to Korean statistical transliteration for information retrieval. *Computer Processing of Oriental Languages.*

K. Jeong, Y. Kwon, and S. H. Myaeng. 1997. 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.*

K. Knight and J. Graehl. 1997. Machine transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics.*

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics.*

Y. C. Park, K. Choi, J. Kim, and Y. Kim. 1996. Development of the data collection ver. 2.0*ktset*2.0 for Korean information retrieval studies. In *Artificial Intelligence Spring Conference.* Korea Information Science Society (*in Korean*).

Frank K. Soong and Eng-Fong Huang. 1991. A tree-trellis based fast search for finding the n best sentence hypotheses in continuous speech recognition. In *IEEE International Conference on Acoustic Speech and Signal Processing*, pages 546–549.

P. Hall and G. Dowling. 1980. Approximate string matching. *Computing Surveys.*