An English to Korean Transliteration Model of Extended Markov Window

SungYoung Jung SungLim Hong* Eunok Paek

Information Technology Lab.
LG Electronics Institute of Technology
Seoul, Korea
e-mail : {syjung, paek}@lg-elite.com

Abstract

Automatic transliteration problem is to transcribe foreign words in one's own alphabet. Machine generated transliteration can be useful in various applications such as indexing in an information retrieval system and pronunciation synthesis in a text-to-speech system. In this paper we present a model for statistical English-to-Korean transliteration that generates transliteration candidates with probability. The model is designed to utilize various information sources by extending a conventional Markov window. Also, an efficient and accurate method for alignment and syllabification of pronunciation units is described. The experimental results show a recall of 0.939 for trained words and 0.875 for untrained words when the best 10 candidates are considered.

Introduction

As the amount of international communication increases, more foreign words are flooding into the Korean language. Especially in the area of computer and information science, it has been reported that 29.4% of index terms are transliterated from or directly written in English in the case of a balanced corpus, KT-SET [18]. The transliteration of foreign words is indispensable in Korean language processing.

In information retrieval, a simple method of processing foreign words is via query term translation based on a synonym dictionary of foreign words and their target transliteration. It is necessary to automate the construction process of a synonym dictionary since its maintenance requires continuous efforts for ever-incoming foreign words. Another area to which transliteration can be applied is a text-to-speech system where orthographic words are transcribed into phonetic symbols. In such applications, maximum likelihood [15], decision tree [1], neural network [10] or weighted finited-state acceptor [19] has been used for finding the best fit.

English-to-Korean transliteration problem is that of generating an appropriate Korean word given an English word. In general, there can be various possible transliterations in Korean which correspond to a single English word. It is common that the newly imported foreign word is transliterated into several possible candidate words based on pronunciation, out of which only a few survive in competition over a period of time. In this respect, a statistical approach makes sense where multiple transliteration variations exist for one word, generating candidates in probable order.

In this paper, we present a statistical method to transliterate English words in Korean alphabet to generate various candidates. In the next section, we describe a phonetic mapping table construction. In Section 2, we describe alignment and syllabification methods, and in Section 3, mathematical formulation for a statistical model is presented. Section 4 provides experimental results, and finally, we state our conclusions.

* *Present address: Sevice Engineering Team, Chollian Service Development Division, DACOM Corporation, Seoul, Korea (E-mail : syrup913@chollian.net)*

1 Phonetic mapping table construction

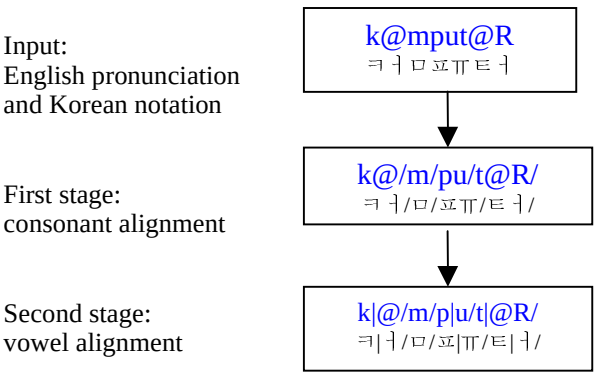
First of all, we generate a mapping between English and Korean phonetic unit pairs (Table 6). In doing so, we use pronunciation symbols for English words (Table 5) as defined in the Oxford computer-usable dictionary [12]. The English and Korean phonetic unit can be a consonant, a vowel or some composite of them so as to make transliteration mapping unique and accurate. The orthography for foreign word transliteration to Korean provides a simple mapping from English to Korean phonetic units. But in reality, there are a lot of transliteration cases that do not follow the orthography. Table 6-1 has been constructed by examining a significant amount of corpus so that we can cover as many cases as possible. Table 6-2 shows complex cases where a combination of two or more English phonemes are mapped to multiple candidates of a composite Korean phonetic unit. This phonetic mapping table is carefully constructed so as to produce a unique candidate in syllabification and alignment in the training stage. When a given English pronunciation can be syllabificated into serveral units or a single composite unit, we adopt a heuristic that only the composite unit consisting of longer phonetic units is considered. For example, the English phonetic unit "u@" can be mapped to a Korean phonetic unit "우어 [u@]" or "워 [w@]" even though the composition of each unit mapping of "u" and "@" can result in other composite mappings such as "유어 [ju@]", "위어 [wI@]", "우여 [wuja]", etc. This composite phonetic unit mapping is also useful for statistical tagging since composite units provide more accurate statistical information when they are well devised.

2 Alignment and syllabification method

The alignment and syllabification process is critical for probabilistic tagging as it is closely linked to computational complexity. There can be combinatorial explosion of state sequences because potential syllables may overlap the same letter sequences. A statistical approach called, Forward-Backward parameter estimation algorithm, is used by Sharman in phonetic transcription problem [2]. But a statistical approach for syllabification requires expensive computational resources and a large amount of training corpus. Moreover, it often results in many improper candidates. In this paper, we propose a simple heuristic alignment and syllabification method that is fast and efficient.

The main principle in separating phonetic units is to manage a phonetic unit of English and that of Korean to be mapped in a unique way. For example, the pronunciation notation "@R" of the suffix "-er" in "computer" is mapped to "어 [@R]" in Korean. In this case, the complex pronunciation "@R" is treated as one phonetic unit. There are many such examples in complex vowels, as in "we" to "웨 [we]", "jo" to "요 [jo]", etc. It is essential to come up with a phonetic unit mapping table that can reduce the time complexity of a tagger and also contribute to accurate transliteration results. Table 6 shows the examples of phonetic units and their mapping to Korean.

The alignment process in training consists of two stages. The first is consonant alignment which identifies corresponding consonant pairs by scanning the English phonetic unit and Korean notation. The second is vowel alignment which separates corresponding vowel pairs within the consonant alignment results of stage 1. Figure 1 shows an alignment example in training. The aligned and syllabificated units are used to extract statistical information from the training corpus. The alignment process always produces one result. This is possible because of the predefined English to Korean phonetic unit mapping in Table 6.



<Figure 1>Alignment example for training data input.
'/' mark: a segmentation position by a consonant
'|' mark: a segmentation position by a vowel.

Figure 1. shows an example of syllabification and alignment. To take the English word "computer" as an example, the English pronunciation notation “k@mpu@R” is retrieved from the Oxford dictionary. In the first stage, it is segmented in front of the consonants “k”, “m”, “p” and “t” which are aligned with the corresponding Korean consonants “ㅋ [k]”, “ㅁ [m]”, “ㅍ [p]” and “ㅌ [t]”. In the second stage, it is segmented in front of the vowels “@”, “u” and “@R” and aligned with the corresponding Korean vowels “ㅓ [@R]”, “ㅠ [ju]” and “ㅓ[@R]”. The composite vowel “@R” is not divided into two simple vowels “@” and “R” since it is aligned to Korean “ㅓ [@R]” in accordance with entry in Table 6-2. When it is possible to syllabificate in more than one ways, only the longest phonetic unit is selected so that an alignment always ends up being unique during the training process.

After the training stage, an input English word must be syllabificated automatically so that it can be transliterated by our tagger. During this stage, all possible syllabification candidates are generated and are given as inputs to the statistical tagger so that the proper Korean notation can be found.

3 Statistical transliteration model

A probabilistic tagger finds the most probable set of Korean notation candidates from the possible syllabificated results of English pronunciation notation. Lee [7, 8, 9] proposed a statistical transliteration model based on the statistical translation model-1 by Brown [2] that uses only a simple information source of a word pair. Various kinds of information sources are involved in the English to Korean transliteration problem. But it is not easy to systematically exploit various information sources by extending the Markov window in a statistical model. The tagging model proposed in this paper exploits not only simple pronunciation unit-to-unit mapping from English to Korean, but also more complex contextual information of multiple units mapping. In what follows, we explain how the contextual information is represented as conditional probabilities.

An English word *E*’s pronunciation *S* is found in a phonetic dictionary. Suppose that S can be segmented into a sequence of syllabificated units $s_1 s_2 \cdots s_n$ where s_i is an English phonetic unit as in Table 6. Also suppose that K is a Korean word, where k_i is the i-th phonetic unit of *K*.

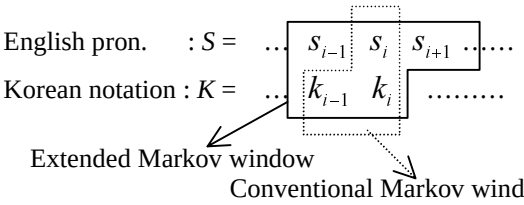
$$S = s_1 s_2 \cdots s_n, \quad K = k_1 k_2 \cdots k_n \tag{1}$$

Let us say *P(E, K)* is the probability that an English word *E* is transliterated to a Korean word *K*. What we have to find is *K* where *P(E, K)* is maximized given *E*. This probability can be approximated by substituting the English word *E* with its pronunciation *S*. Thus, the following formula holds.

$$\begin{aligned} &\arg\max_K P(E, K) \\ &\cong \arg\max_K P(S, K) \\ &= \arg\max_K P(K \mid S) P(S) \end{aligned} \tag{2}$$

where *P(S)* is called *language model* and *P(K|S)* is called *translation model*. *P(S)* is not constant given a fixed input word because there can be a number of syllabification candidates.

In determining k_i, four neighborhood variables are taken into account, while conventional tagging models use only two neighborhood variables. The extended Markov window of information source is defined as in Figure 2. It also shows a conventional Markov window using a dashed line. Mathematical formulation for Markov window extension is not an easy problem since extended window aggravates data sparseness. We will explain our solution in the next step.



<Figure 2> Extended Markov window of information source for k_i

Now, the translation model, *P(K|S)* in equation (2) can be approximated by Markov assumption as follows.

$$P(K \mid S) \cong \prod_i P(k_i \mid k_{i-1}, s_{i-1}, s_{i,} s_{i+1}) \qquad (3)$$

Equation(3) still has data sparseness problem in gathering information directly from the training corpus. So we expand it using Markov chain in order to replace the conditional probability term in (3) with more fragmented probability terms.

$$P(k_i \mid k_{i-1}s_{i-1}s_i s_{i+1}) = \frac{P(k_i k_{i-1}s_{i-1}s_i s_{i+1})}{P(k_{i-1}s_{i-1}s_i s_{i+1})}$$
$$= \frac{P(k_{i-1})P(k_i \mid k_{i-1})P(s_{i-1} \mid k_{i-1}k_i)P(s_i \mid k_{i-1}k_i s_{i-1})P(s_{i+1} \mid k_{i-1}s_{i-1}k_i s_i)}{P(k_{i-1})P(s_{i-1} \mid k_{i-1})P(s_i \mid k_{i-1}s_{i-1})P(s_{i+1} \mid k_{i-1}s_{i-1}s_i)}$$
$$\cong P(k_i \mid k_{i-1}) \frac{P(s_{i-1} \mid k_{i-1}k_i)}{P(s_{i-1} \mid k_{i-1})} \cdot \frac{P(s_i \mid k_i s_{i-1})}{P(s_i \mid s_{i-1})} \cdot \frac{P(s_{i+1} \mid k_i s_i)}{P(s_{i+1} \mid s_i)}$$
(4)

In Equation (4), there are two kinds of approximations in probability terms. First, $P(s_i \mid k_i s_{i-1})$ and $P(s_i \mid s_{i-1})$ are substituted for $P(s_i \mid k_{i-1}k_i s_{i-1})$ and $P(s_i \mid k_{i-1}s_{i-1})$, respectively. This approximation is based on our heuristic that k_{i-1} and s_{i-1} provide somewhat redundant information in determining s_i. Secondly, $P(s_{i+1} \mid k_i s_i)$ and $P(s_{i+1} \mid s_i)$ are substituted for $P(s_{i+1} \mid k_{i-1}s_{i-1}k_i s_i)$ and $P(s_{i+1} \mid k_{i-1}s_{i-1}s_i)$, respectively, based on a heuristic that $k_{i-1}s_{i-1}$ is farther off than $k_i s_i$, and is redundant. Equation (4) can be reduced to Equation (5) because $\frac{P(s_{i-1} \mid k_{i-1}k_i)}{P(s_{i-1} \mid k_{i-1})}$ of (4) is equivalent to $\frac{P(k_i \mid s_{i-1}k_{i-1})}{P(k_i \mid k_{i-1})}$ mathematically.

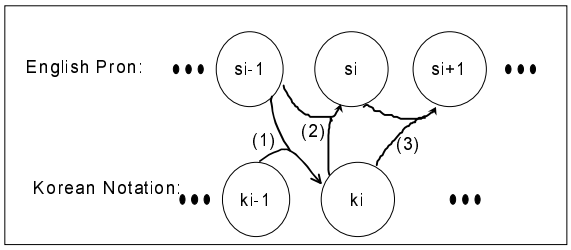
$$P(k_i \mid k_{i-1}s_{i-1}s_i s_{i+1})$$
$$\cong \frac{P(k_i \mid s_{i-1}k_{i-1})P(s_i \mid k_i s_{i-1})P(s_{i+1} \mid k_i s_i)}{P(s_i \mid s_{i-1})P(s_{i+1} \mid s_i)} \qquad (5)$$

The language model we use is a bigram language model (Brown et al. [6])

$$P(S) \cong \prod_i P(s_i \mid s_{i-1}) \qquad (6)$$

Now, our statistical tagging model can be formulated as Equation (7) when the translation model (5) and the language model (6) are applied to the transliteration model (2)

$$\therefore \arg\max_K P(S, K)$$
$$\cong \arg\max_K \prod_i \frac{P(k_i \mid s_{i-1}k_{i-1})P(s_i \mid k_i s_{i-1})P(s_{i+1} \mid k_i s_i)}{P(s_{i+1} \mid s_i)} \qquad (7)$$



<Figure 3> Statistical information source used in the extended Markov window

(1) $P(k_i \mid s_{i-1}k_{i-1})$ (2) $P(s_i \mid k_i s_{i-1})$ (3) $P(s_{i+1} \mid k_i s_i)$

Figure 3 pictorially summarizes the final information sources that our statistical tagger utilizes. It can be thought of as a generalized case of prevalent Part-of-Speech tagging model. When $P(k_i \mid s_{i-1}k_{i-1})$ is approximated as $P(k_i \mid k_{i-1})$, $P(s_i \mid k_i s_{i-1})$ as $P(s_i \mid k_i)$, and $P(s_{i+1} \mid k_i s_i)$ as $P(s_{i+1} \mid s_i)$, then Equation (7) is reduced to a conventional bigram tagging model (Eq. 8), that is a base model of Brown model-1 [2], Charniak [4], Merialdo [11] and Lee [7, 8, 9].

$$\arg\max_K P(S, K) \cong \arg\max_K \prod_i P(k_i \mid k_{i-1})P(s_i \mid k_i) \qquad (8)$$

Equation (7) is the final tagging model proposed in this paper. We use a back-off strategy [10, 11] as follows, because our tagging model may have a data sparseness problem.

$$P(k_i \mid s_{i-1}k_{i-1}) \approx P(k_i \mid k_{i-1}), \text{ if } Count(s_{i-1}k_{i-1}) = 0$$
$$P(s_i \mid k_i s_{i-1}) \approx P(s_i \mid k_i), \text{ if } Count(k_i s_{i-1}) = 0$$
$$P(s_{i+1} \mid k_i s_i) \approx P(s_{i+1} \mid s_i), \text{ if } Count(k_i s_i) = 0$$

Each probability term in equation (7) is obtained from the training data. The statistical tagger modeled here uses Viterbi algorithm [12] for its search to get N-Best candidates.

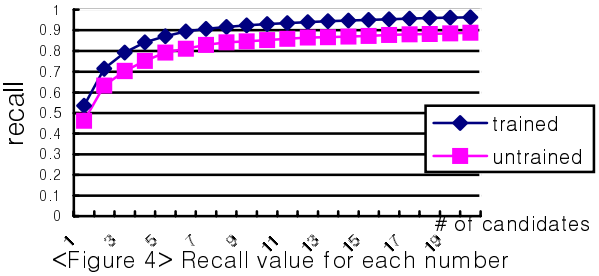
4. Experimental results

For the evaluation we constructed a training corpus of 8368 English-Korean word pairs. One English word can have one or more Korean transliteration entries in the corpus. 90% of the corpus is used as training data and 10% of the corpus as test data. For more objective experiment evaluation, we estimated word-level accuracy based on exact string match even though many other papers are based on lexical-level distance to the correct word. We adopted a recall measure based on word-level accuracy. Recall measure is the average number of generated correct words divided by the total word count of prepared correct answer set given an input word (Eq. 9). Precision measure is the average number of retrieved correct words divided by the number of generated candidates (Eq. 10).

$$\text{Recall} = \frac{count(generated\_correct\_words)}{count(possible\_correct\_words)} \quad (9)$$

$$\text{Precision} = \frac{count(generated\_correct\_words)}{count(generated\_words)} \quad (10)$$

For words not found in the pronunciation dictionary, a transcription automata is used to transform the English alphabet to the Korean alphabet. A transcription automata can be helpful because it uses alphabetic information that our statistical tagger does not use. The automata produces one result and attaches it at the end of N-best results of the statistical tagger. This automata has about 500 transcription rules, based on previous, current, and next context window and production alphabet.



<Figure 4> Recall value for each number of candidates

All experimental results are estimated by 10-fold cross validation for more accurate results. Table 1 shows the estimated recall values for the 10-best results generated by the tagger and for the case when transcription automata used as well. Figure 4 shows recall values given a number of candidates.

	Trained	Test
Pure statistical tagger (eq. 7)	0.925	0.850
Transcription automata used as well	0.939	0.875

<Table 1> estimated recall result on 10-best results

We estimated recall values in the same environment for conventional tagging model (Eq. 8) in order to compare accuracy improvement by the Extended Markov window model (Eq. 7) without transcription automata (Table 2).

	Trained	untrained
Extended Markov window model (Eq. 7)	0.925	0.850
Conventional tagging model (Eq. 8)	0.878	0.796

<Table 2> comparison of recall values on 10-best results of the proposed Extended Markov model with conventional tagging model

It cannot be compared directly with the results of other models since other related works are on a different domain and they adopt different evaluation measures such as lexical-level accuracy [1, 7, 10]. There is a model that is in the same domain, i.e., English-to-Korean transliteration [1, 8, 9], but it adopts a lexical-level accuracy measure [7, 9], or a subjective evaluation measure such as human judgment [8, 9]. Table 3 shows the comparison with Lee's model which adopted the average value of trained and untrained results, even though the average of trained and untrained results makes no sense since a registered dictionary lookup method can make all experiments on trained data 100% accurate. Accuracy measure on untrained data should be the measure for comparison among different experiments for fairness.

Extended Markov window model	Trained	Untrained
	0.962	0.888
Lee – hybrid [17]	Average:(train+untrained) / 2	
	0.471	

<Table 3> Comparison with Lee's model, word based recall measure on 20-best results. Lee's result of single frequency coverage [17] is the same as a recall measure in this paper.

Lee's model is a fully statistical approach even in pronunciation unit alignment and syllabification that may cause inaccurate results, while we use a heuristic approach in pronunciation unit alignment. Another significant difference is that Lee's model uses only conventional information sources such as a bigram while our model use various information sources from extended Markov window. Lee's model transliterates using English alphabet directly without pronounciation dictionary so that it can be better for unknown words or proper noun.

	Trained	Untrained
Extended Markov window	64.0%	54.9%
transcription automata	(no training)	36.4%
MBRtalk [16]	(100%)	43%
Neural Net [10]	37.7%	26.2%
WFSA [19]	?	64%
Lee's – modified Direct [9]	Average:(train + untrained) / 2	
	38.1%	

<Table 4> comparison with other models – word accuracy (precision) of 1-best candidate.

Table 4 shows the comparison with MBRtalk [16], Neural Network [10], Weighted finite-state acceptor (WFSA) [19] and direct transliteration [9] even though they are based on different problem domains. MBRtalk and Neural Network models are based on English word's pronunciation generation. WFSA is for English-to-Japanese transliteration. Experiment for training data in MBRtalk makes no sense, since it finds the most similar word in a database that stores all the training data; thus the result would always produce the exact answer. The results show that the model we propose indicates the good performance.

Conclusion

We have proposed a statistical English-to-Korean transliteration model that exploits various information sources. This model is a generalized model from a conventional statistical tagging model by extending Markov window with some mathematical approximation techniques. An alignment and syllabification method instead of a statistical method is proposed for accurate and fast operation. The experimental results show that the model proposed in this paper demonstrates significant improvement in its performance.

Phonetic symbol	example	Phonetic symbol	example
i	b<u>ea</u>d	S	<u>sh</u>ed
I	b<u>i</u>d	Z	bei<u>ge</u>
e	b<u>e</u>d	tS	e<u>tch</u>
&	b<u>a</u>d	dZ	e<u>dge</u>
A	b<u>ar</u>d	p t k b d g	
0 (zero)	c<u>o</u>d	m n f v s z	
O (cap O)	c<u>or</u>d	r l w h j	
U	g<u>oo</u>d	ei	d<u>ay</u>
u	f<u>oo</u>d	@U	g<u>o</u>
V	b<u>u</u>d	aI	<u>eye</u>
3	b<u>ir</u>d	aU	c<u>ow</u>
@	ab<u>ou</u>t	oI	b<u>oy</u>
N	si<u>ng</u>	I@	b<u>ee</u>r
T	<u>th</u>in	e@	b<u>are</u>
D	<u>th</u>en	U@	t<u>our</u>

<Table 5: Pronunciation symbol to Ascii mapping [3]>

English phonetic unit	Korean unit (orthography)	Korean unit (extra)
N	ㅇ	
p	ㅍ ㅂ ㅃ	ㅃ
t	ㅌ ㅅ ㅌ	ㅆ ㄸ ㅊ ㄷ
k	ㅋ ㄱ ㅋ	ㄲ ㅎ
s	ㅅ ㅅ	ㅈ ㅆ ㅉ
d	ㄷ ㄷ	ㅌ ㅅ ㅈ
...	...	...
l	ㄹ ㄹㄹ	
w	(ㅇ)	ㅎ
@	어	오 이 우 으 에 아 여 애 예 워 야 요 유
u	우	유 오 으
a	아	와 (fa 화) 야 오
e	에	예 웨 이 어 으 아
o	오	요
...	...	...

<Table 6-1: Examples of English to Korean phonetic unit mapping (simple cases)>

English Phonetic unit	Korean units (orthography)	Korean units (extra)
I@	이어	이우 이오 이요 유
@U	오	오우 요
U@	우어 워	
@R		어
AI	아이	이
eI	에이	아 에
wa	와	(화) 우아 워
w3	웨	워
we	웨	워
w@	워	와 워 웨
w@U	워	
ja	여	요 어 유 이아 야 이어
jA	야	아
...	...	...

<Table 6-2: Examples of English to Korean phonetic unit mapping (complex cases)>

References

[1] Bahl, L. R. and et al. (1991) *Decision Trees for Phonological Rules in Continuous Speech.* IEEE ICASSP, pp. 125-138.

[2] Brown, P. F. and et al. (1990) *The Mathematics of Statistical Machine Translation: Parameter Estimation.* Computational Linguistics, Computational Linguistics, V. 19, No. 2, June.

[3] Brown, P. F. and et al. (1992) *Class-Based n-gram Models of Natural Language*. Association for Computational Linguistics, V. 18, No. 4.

[4] Charniak, E. (1993) *Statistical Models in Natural-Language Processing.* MIT Press.

[5] Jung, S. Y. and et al. (1996) *Markov random field based English Part-of-Speech tagging system.* International conference of Computational Linguistics (COLING).

[6] Katz, S. (1987) *Estimation of probabilities from sparse data for the language model component of a speech recognizer*. IEEE Transactions on ASSP, 34(3), pp400-401.

[7] Lee, J. S. and K. S. Choi (1997) *Automatic Foreign Word Transliteration Model for Information Retrieval.* 4th Korean Information management conference proceeding, Seoul, Korea, pp17 –24.

[8] Lee, J. S. and K. S. Choi. (1997) *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 (IRAL'97), pp123-128, Tsukuba-shi, Japan.

[9] Lee, J. S. and K. S. Choi. (1998) *English to Korean Statistical Transliteration for Information Retrieval.* Computer Processing of Oriental languages.

[10] Lucas, S.M. and R.I. Damper. (1991) *Syntactic neural networks for bi-directional text-phonetics translation.* Talking Machines, North Holland, pp127-141.

[11] Merialdo, B. (1994) *Tagging English Text with a Probabilistic Model.* Association for Computational Linguistics, V. 20, No. 2, pp155-171.

[12] Roger, M. (1992) *A Description of a computer-usable dictionary file based on the Oxford advanced learner's dictionary of current English.* Department of computer science, University of London.

[13] Rosenfeld, R. (1994) *Adaptive Statistical Language Modelling: A Maximum Entropy Approach.* Technical report, CMU-CS-94-138, Carnegie Mellon University.

[14] Salton, G. (1988) *Automatic Text Processing.* Addison Wesley.

[15] Sharman, R. A. (1994) *Syllable-based Phonetic Transcription by Maximum Likelihood Methods.* Computational Linguistics, pp. 1279.

[16] Stanfill, C. (1986) *Toward Memory-Based Reasoning.* Communication of the ACM, vol. 29, no. 12, pp1213-1228, December.

[17] Viterbi, A. J. (1967) *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.* IEEE Transactions on Information Theory, v. IT-13, no. 2, pp 260-269, April.

[18] Kim, J., Y. Kim and S. Kim (1994) *Development of the Data collection (KTSET) for Korean Information Retrieval Studies.* The 6th Hangul and Korean Information Processing, pp. 378-385.

[19] Knight, Kevin and Jonathan Graehl (1997) *Machine Transliteration*, Proc. 35th Mtg, Assoc. for Computational Linguistics