An English-Korean Transliteration Model Using Pronunciation and Contextual Rules

Jong-Hoon Oh, and Key-Sun Choi
Computer Science Division, Dept. of EECS, Korea Advanced Institute of Science & Technology (KAIST) / Korea Terminology Research Center for Language and Knowledge Engineering (KORTERM), 373-1, Kusong-dong, Yusong-gu, Taejon, 305-701, Korea
Email: {rovellia,kschoi}@world.kaist.ac.kr

Abstract

There is increasing concern about English-Korean (E-K) transliteration recently. In the previous works, direct converting methods from English alphabets to Korean alphabets were a main research topic. In this paper, we present an E-K transliteration model using pronunciation and contextual rules. Unlike the previous works, our method uses phonetic information such as phoneme and its context. We also use word formation information such as English words of Greek origin. With them, our method shows significant performance increase about 31% in word accuracy.

1.Introduction

In Korean, many technical terms in a domain specific text, especially science and engineering are from foreign origin. Sometimes they are written in their original forms and sometimes they are transliterated into Korean words in various forms. This makes difficult to handle them in natural language processing. Especially information retrieval, words with the same meanings are treated as different ones because of their different forms.

One possible solution can be a dictionary, which contains English words and their possible transliterated forms. However, this is not a practical solution because technical terms, which mainly cause the problem, usually have rich productivity. The other solution can be automatic transliteration. There have been works on automatic transliteration from English to other languages – English to Japanese (Kang *et al*., 1996; Knight *et al*., 1997), and English to Korean (Kang *et al*., 2000; Kang *et al*., 2001; Kim *et al*., 1999; Lee *et al*., 1998).

In E-K transliteration, direct converting methods from English alphabet to Korean alphabet were a main research topic (Kang *et al*., 2000; Kang *et al*., 2001; Kim *et al*., 1999; Lee *et al*., 1998). In the works, machine learning techniques such as a decision tree and a neural network were used.

However, transliteration is more phonetic process than orthographic process: 'h' in the Johnson does not make any Korean character (Knight *et al*., 1997). Therefore, patterns for E-K transliteration acquired from English/Korean alphabets as in the previous works, may not be effective. In the previous works, they did not consider origin of English – pure English (*e.g.,* board), English words with Greek origin (*e.g.,* hernia) and so on In E-K transliteration, origin of English words determine the way of transliteration. Our method uses phonetic information such as phoneme and its context as well as orthography. English words of Greek origin are also considered in transliteration.

This paper organized as follows. In section 2, we survey related works. In section 3, we will describe the details of our method. In section 4, the results of experiments are represented. Finally, the conclusion follows in section 5.

2. Related works

2.1 Probability based transliteration

(Lee *et al*., 1998) used formula (1) to generate a transliterated Korean word 'K' for a given English word 'E'. Lee *et al*. (1998) defined a pronunciation unit. It is a chunk of graphemes or alphabets that can be mapped to phoneme. They divided an English word into pronunciation units

(PUs) for transliteration. For example, an English word ‘board (/B AO R D/)’ can be divided into ‘b/B/**:** oa/AO/**:** r/R/**:** d/D/’[1] – ‘b’, ‘oa’, ‘r’ and ‘d’ are PUs. An English word ‘E’ was represented as ‘$E=epu_1,epu_2,\ldots,epu_n$’ where epu_i was the i^{th} PU. Sequences of Korean PUs, $K_1,K_2,\ldots,K_m$, where ‘$K_i= kpu_{i1},kpu_{i2},\ldots,kpu_{in}$’ were generated according to epu_i. Lee *et al*. (1998) considered all possible English PU sequences and corresponding Korean PU sequences for a given English word, because its pronunciation was not determined. For example, ‘data’ can have PU sequences such as ‘d **:**at **:**a’, ‘da **:**ta’, ‘d **:**a **:**t **:**a’ and so on. If the total number of English PU in E is N and the average number of kpu_i generated by epu_i is M, the total number of generated Korean PU sequences will be about N*M. Then he selected the best result among them as a Korean transliteration word.

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

$$P(K) \cong p(kpu_1)\prod_{i=2}^{n} p(kpu_i \mid kpu_{i-1}) \quad (2)$$

$$P(E \mid K) \cong \prod_{i=1}^{n} p(epu_i \mid kpu_i) \quad (3)$$

Kim *et al.*, (1999) used the same formula as Lee’s (1998) except $P(E|K)$ (formula(4)). He used additional information – Korean PUs kpu_{i-1} and kpu_{i+1} – and used a neural network to approximate $P(E|K)$.

$$P(E \mid K) \cong \prod_{i=1}^{n} p(epu_i \mid kpu_{i-1}, kpu_i, kpu_{i+1}) \quad (4)$$

Probability based transliteration showed about 40% precision on E-K transliteration with 1,500 E-K pairs for training and 150 E-K pairs for testing.

2.2 Decision Tree based transliteration

Kang, *et al*. (2000; 2001) proposed an English alphabet-to-Korean alphabet conversion method based on a decision tree. This method used six attribute values – left three English alphabets and right three English alphabets – for determining Korean alphabets corresponding to English alphabets. For each English alphabet, its corresponding decision trees are constructed. Table 1 shows an example of transliteration for an English word ‘data’. In table 1, (E) represents a current English alphabet, K represents generated Korean alphabets by decision trees.

L3	L2	L1	(E)	R1	R2	R3		K
<	<	<	d	a	t	a	→	‘d’
<	<	d	a	t	a	>	→	‘e-i’
<	d	a	t	a	>	>	→	‘t’
d	a	t	a	>	>	>	→	‘a’

Table 1. An example of decision tree based transliteration

This method showed about 49% precision for 6,185 E-K pairs for training and 1,000 E-K pairs for testing.

Though the previous works showed relatively good results, they also showed some limitations. Because they focused on a converting method from English alphabet to Korean alphabet, they did not consider phonetic features such as phoneme and word formation features such as origin of English. This makes some errors when pronunciation and origin of English were important clues for transliteration - ‘Mcdonald’ (pronunciation is needed) and ‘amylase’ (origin of English word is needed).

3. An English-Korean Transliteration Model using Pronunciation and Contextual Rules

3.1 Overall System Description

Figure1 shows the overall system description. Our method is composed of two phases – alignment (section 3.2) and transliteration (section 3.3, 3.4, 3.5 and 3.6).

First an English pronunciation unit[2] (*hearafter*, EPU) and its corresponding phoneme are aligned. EPU-to-Phoneme alignment is to find out the most phonetically probable correspondence between an English pronunciation unit and phoneme. EPU to phoneme aligned results acquired from the alignment algorithm offer training data for estimating pronunciation of English words, which are not registered in a pronunciation dictionary, for example ‘zinkenite’. Second, English words are transliterated into Korean words through several steps. Using an English

[1] Henthforce, ‘**:**’ will be used as a PU boundary

[2] The term ‘pronunciation unit’ will be used as the same meaning as in the Lee’s (Lee et al., 1998)

pronunciation dictionary (P-DIC), we can assign pronunciation to a given English word. When it is not registered in P-DIC, we investigate that it has a complex word form (section 3.3). For detecting a complex word form, we divide a given English word into two words (word+word)[3] using entries of P-DIC. If both of them are in P-DIC, we can assign pronunciation to the given word otherwise we should estimate pronunciation (section 3.5). Then, we check whether the English word is from Greek origin or not (section 3.4). Because a way of E-K transliteration for the English words of Greek origin is different from that for pure English words, it is important to detect them. Pronunciation for English words, which are not registered in a P-DIC, is estimated (section 3.5) in the next step. Finally, Korean transliterated words are generated using conversion rules (section 3.6). The right side of figure 1 shows a transliteration example for an English word, ‘cutline’.

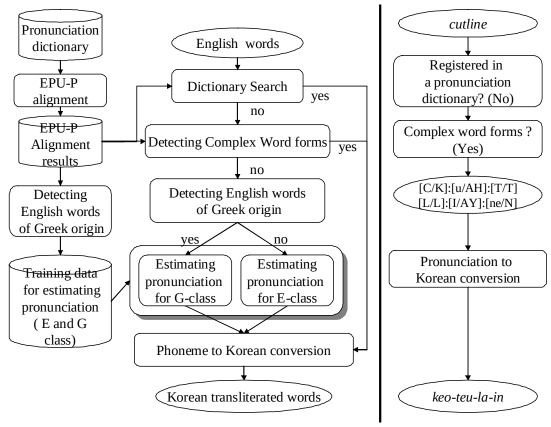

Fig. 1 Overall system description

3.2 EPU-to-Phoneme Alignment

EPU-to-Phoneme (hereafter, EPU-P) alignment is to find out the most phonetically probable correspondence between an English pronunciation unit and phoneme. For example, one of the possible alignment for an English word ‘board’ and its pronunciation ‘/B AO R D/’[4] is as follows.

English	b	oa	r	d
	|	|	|	|
Pronunciation	/B/	/AO/	/R/	/D/

Table 2. One possible alignment between English word ‘board’ and its pronunciation

For automatic EPU-P alignment, we used the modified version of Kang’s E-K alignment algorithm (Kang *et al*., 2000; Kang *et al*., 2001). It is based on Covington’s algorithm (Covington, 1996). Covington views an alignment as a way of stepping through two words – a word in one side and a word in the other side – while performing ‘*match*’ or ‘*skip*’ operation on each step. Kang added ‘*forward bind*’ and ‘*backward bind*’ operations to consider one-to-many, many-to- one and many-to-many alignments

Operation	Condition	Penalty
Match	Similar C/CP	0
	V/VP	0
	V/SVP or C/SVP	30
	Dissimilar C/CP	240
	V/CP or C/VP	250
Bind	Similar C/CP	0
	V/VP	0
	V/SVP or C/SVP	30
	Dissimilar C/CP	190
	V/CP or C/VP	200

Table 3. Penalty metrics: C, V, CP, VP, and SVP represent consonants, vowels, consonant phonemes, vowel phonemes [5] and semi-vowel phonemes respectively.

English	b	o	a	r	D	Total
Operation	M	M	<	M	M	Penalty
Pronunciation	B	AO	<	R	D	
Penalty	+0	+0	+0	+0	+0	0

Table 4. The best alignment result for an English word ‘board’. ‘M’ represents ‘match’, and ‘<’ represents ‘backward bind’.

Unlike the previous alignment algorithm, we combine *‘skip’* and *‘bind’* operations because the *‘skip’* operation can be replaced with the *‘bind’* operation. This makes all PUs to be mapped into phoneme. It means that our algorithm does not allow null-to-phoneme alignment or PU-to-null alignment. All the valid alignments that are possible by ‘*match*’, and ‘*bind*’ operations can be generated. Alignment

---

3 ‘broadcasting’ may be divided into three words : ‘broad’, ‘cast’ and ‘ing’. But from the training corpus and pronunciation dictionary, all of complex word is divided into two words like ‘broad’ and ‘casting’.

4 (*www.cs.cmu.edu/~laura/pages/arpabet.ps*): ARPAbet symbol will be used for representing phonemes. ARPAbet is one of the method for coding phonemes into ASCII chracters.

5 In this paper, vowel pronunciation includes diphthongs.

may be interpreted as finding the best result among them. To find the best result, a penalty scheme is used – the best alignment result is one that has the least penalty values. Since Kang's method focused on an E-K character alignment, a penalty scheme and an E-K character-matching table were restricted to an E-K alignment. Instead of Kang's E-K character penalty scheme, we developed an EPU-P penalty scheme and an EPU-P matching table using manually aligned EPU-P data. We assume that all vowels can be aligned with all vowel phonemes without penalty. Table 3 shows our penalty metrics and table 4 shows an example of EPU-P alignment.

We aligned about 120,000 English word and Pronunciation pairs in 'The CMU Pronouncing Dictionary'. For evaluating performance of the alignment, we randomly selected 100 results. The performnance of EPU-P alignment is 99%.

3.3 Dealing with a Complex word form

Some English words are not in P-DIC, because they are in a complex word form. In this paper, we define words in a complex word form as those composed of two base nouns in P-DIC. When a given word is not in P-DIC, it is segmented into all possible two words. If the two words are in P-DIC, we can assign their pronunciation. For example, 'cutline' can be segmented into 'c+utline', 'cu+tline', 'cut+line' and so on. 'cut+line' is the correct segmentation of 'cutline', because 'cut' and 'line' are in the P-DIC. If words are not in P-DIC and they are not in a complex word form, we should estimate their pronunciation. The details of estimating pronunciation will be described in the section 3.5.

3.4 Detecting English words of Greek origin

In Korean, there are two methods for E-K transliteration – 'written word transliteration' and 'spoken word transliteration' (Lee *et al*., 1998). The two methods use similar mechanism for consonant transliteration. However, 'written word transliteration' uses its character and 'spoken word transliteration' uses its phoneme when they transliterate vowels. For example, 'a' in 'piano' can be transliterated into 'pi-***a***-no' with its character and 'pi-***e***-no' with its phoneme. Since, a vowel in a pure English word is usually transliterated using its phoneme and that in an English word of Greek origin is usually transliterated with its character in E-K transliteration- for example, 'hernia' (he-reu-ni-a), 'acacia' (a-ka-si-a), 'adenoid' (a-de-no-i-deu) and so on -, it is important to detect them. We use suffix and prefix patterns for detecting English words of Greek origin (Luschnig, 2001)[6] and table 5[7] shows the patterns. If words have the affixes in table 5, we determine them as words of Greek origin otherwise pure English words.

Suffix	-ic, -tic, -ac, -ics, -ical, -oid, -ite, -ast, -isk, -iscus, -ia, -sis, -me, -ma
Prefix	amphi-, ana-, anti-, apo-, dia-, dys-, ec-, ecto-, enantio-, endo-, epi-, cata-, cat-, meta-, met-, palin-, pali-, para-, par-, peri-, pros-, hyper-, hypo-, hyp-

Table 5. Suffix and prefix patterns for detecting English words of Greek origin.

3.5 Estimating Pronunciation

Estimating pronunciation is composed of two steps. Using aligned EPU-P pairs as training data, we can find EPUs in the given English word (Chunking EPU) and assign their appropriate phoneme (EPU-to-Phoneme assignment). For dealing with English words of Greek origin, we categorize EPU-P aligned data into pure English words (E-class) and English words of Greek origin (G-class). Then we construct the 'Chunking EPU' module and the 'EPU-to-Phoneme assignment' module for each class.

'Chunking EPU' is to find out boundaries of EPUs in English words. For example, we can find EPUs in 'board' as 'b:oa:r:d'. For chunking EPU, we used C4.5 (Quilan, 1993) with ten attributes – left five alphabets and right five alphabets and the setting shows the best result among various settings such as eight attributes (left four and right four - 87.2%) and so on. [8].

[6] 38 Grek affixes out of 249 Latin and Greek affixes in 120 categories described in (John, 1953) are used. 63 out of the 120 categories share the meaning though their form is somewhat different

[7] In this paper, some Greek affixes are not used, because they such as prefix 'a-', 'an-', and postfix '-y', '-m' may cause error.

[8] C4.5 is one of the popular method for recognizing boundary of chunks. Unlike Kang *et al*., (2000)'s method,

We use 90% of EPU-P aligned data as training data and 10% of those as test data. Our 'Chunking EPU' module shows 91.7% precision.

$$\arg\max_P p(P \mid E) = \arg\max_P p(P)\, p(E \mid P) \quad (5)$$

$$P(P) \cong p(p_1) \prod_{i=2}^{n} p(p_i \mid p_{i-1}) \quad (6)$$

$$P(E \mid P) \cong \prod_{i=1}^{n} p(epu_i \mid p_i) \quad (7)$$

Then we can assign phoneme to each EPU. For the given EPU sequence '$E=epu_1,epu_2,\ldots,epu_n$' and its possible phoneme sequences $P_1,..,P_m$ where '$P_i=p_{i1},p_{i2},\ldots,p_{in}$', the 'EPU-to-Phoneme assignment' task is to find out the most probable phomene sequence '$P_i=p_{i1},p_{i2},\ldots,p_{in}$'. It can be represented as formula (5). $p(P)$ and $p(E|P)$ are approximated as formula (6) and (7).

3.6 Phoneme-to-Korean Conversion

Our Phoneme-to-Korean (P-K) conversion method is based on English-to-Korean Standard Conversion Rule (EKSCR) (Ministry, 1995). EKSCR is composed of nine general rules and five rules for specific cases – each rule contains several sub-rules. It describes a transliteration method from English alphabets or phonemes to Korean alphabets. It uses English phoneme as a transliteration condition – if a phoneme is A then transliterate into a Korean alphabet B. However, EKSCR does not contain enough rules to generate correct Korean words for corresponding English words, because it mainly focuses on a way of mapping from one English phoneme to one Korean character without context of phonemes and PUs. For example, an English word 'board' and its pronunciation '/B AO R D/', are transliterated into 'bo-reu-deu' by EKSCR – the correct transliteration is 'bo-deu'. In E-K transliteration, the phoneme 'R' before consonant phonemes and after vowel phonemes is rarely transliterated into Korean characters (Note that the phoneme 'R' in English words of Greek origin is transliterated into a Korean consonant 'r' frequently.) These contextual rules are very important to generate correct Korean transliterated words.

We capture contextual rules by observing errors in the results, which are generated by applying EKSCR to 200 randomly selected words from the CMU pronunciation dictionary. The selected words are not in the test data in the experiment. Among the generated rules, we selected 27 contextual rules with high frequency (above 5). Table 6 shows some rules and their conditions in which rules will be fired. There are three conditions – 'Context', 'TPU (Target PU)', and 'TP (Target Phoneme)'. In context condition, '[]', '{}', C, VP, and CP represent phoneme, pronunciation unit, consonant, vowel phonemes and consonant phonemes respectively. The rule with context condition, '[R] after VP and before CP', is not fired for the English words of Greek origin. Except it, all rules are applied to both classes (E-class and G-class).

Condition			Korean Characters
Context	TPU	TP	
C+ {le}	'le'	AH L	'eul'
{or} in the end of a word	'or'	ER	'eo'
{or} in a word	'or'	ER	'eu'
{sm} in the end of a word	'sm'	S AH M	'jeum'
[R] after VP and before CP	'r'	'R'	'eu'

Table 6. Some contextual rules

4.Experiment

4.1 Experimental Setup

We use two data sets for an accuracy test. *Test Set I* (Lee *et al.*, 1998) is composed of 1,650 E-K pairs. Since, the test set was used as a common testbed for (Lee *et al.*, 1998; Kim *et al.*, 1999; Kang *et al.*, 2000; Kang *et al.*, 2001), we use them as a testbed for comparison between our method and other methods. For comparison, 1,500 pairs are used as training data for other methods and 150 pairs are used as test data for our method and other methods. *Test set II* (Kang *et al.*, 2000) consists of 7,185 E-K pairs – the number of training data is 6,185 and that of test data is 1,000. We use *Test set II* to compare our

our method produces EPU and it phoneme. This makes possible for a E-K conversion method (in section 3.6) to use context of EPU and its phoneme. Because an alphabet-to-alphabet mapping method did not use EPU and its phoneme, it may show some errors when phoneme and its context are the most importnat clues, for example, 'Mcdonald'.

method with (Kang *et al.*, 2000), which shows the best result among the previous works.

Evaluation is performed by word accuracy (W. A.) and character accuracy (C.A.), which were used as the evaluation measure in the previous works (Lee and Choi 1998; Kim and Choi 1999; Kang and Choi 2000).

$$W.A. = \frac{\#of\ correct\ words}{\#of\ generated\ words} \quad (8)$$

$$C.A. = \frac{L-(i+d+s)}{L} \quad (9)$$

where L represents the length of the original string, and i, d, and s represent the number of insertion, deletion and substitution respectively. If $L < (i+d+s)$, we consider it as zero (Hall and Dowling, 1980).

We perform the three experiments as follows.

- *Comparison Test:* Comparison between our method and the previous works
- *Dictionary Test*: Performance of transliteration for words in a pronunciation dictionary and that for others
- *Component Test*: Effectiveness of each component

4.2 Experimental results

4.2.1 Comparison Test

Method	C.A	W.A
[Lee *et al.*, 1998]	69.3%	40.7%[9]
[Kim *et al.*, 1999]	79.0%	35.1%
[Kang *et al.*, 2000]	78.1%	37.6%
Our method	90.82%	56.0%

Table 7 Comparison test results for *Test set I*

Method	C.A	W.A
[Kang *et al.*, 2000, 2001]	81.8%	48.7%
Our method	92.86%	63.0%

Table 8 Comparison test results for *Test set II*.

Table 7 and 8 show results of comparison test for *Test set I* and *Test set II* respectively. In the tables our method shows higher performance especially in W.A. Moreover, our method shows higher performance in C.A. It means that the generated words by our method are more similar to the correct transliteration, when they are not the correct answer.

4.2.2 Dictionary Test

For the dictionary test, we use test data of *Test set II*. In the result, 'registered' words show higher performance. It can be analysed that contextual rules are constructed using registered words in a P-DIC and estimating pronunciation module makes some errors. However, 'not registered' words also show relatively good performance.

	C.A	W.A	# of words
Registered	93.49%	67.83%	687
Not registered	91.47%	52.40%	313

Table 9. Dictionary test results.

4.2.3 Component Test

For the component test, we use words, which are 'not registered' in *Test set II*. Components, which are tested in 'Component test' are 'Dealing with words in a complex word form'[C], 'Detecting English words of Greek origin' [G], and 'Contextual rules' [R]. In the result, [G] and [R] show good results in contrary to [C]. There are so few words in complex word forms that [C] does not show significant performance improvement though the performance is relatively good –about 70% W.A. for 43 words (43 words out of total 313 words).

For the effective comparison, it will be necessary to consider the number of words, which each component handles. Our method shows better performance than 'W/O [R]'(EKSCR). It indicates that contextual rules are important.

Method	C.A.	W.A.
W/O [C], [G], and [R]	87.90%	23.96%
W/O [G]	88.45%	36.10%
W/O [C]	91.99%	50.16%
W/O [R]	89.78%	44.41%
[C]+[G]+[R] (proposed)	91.47%	52.40%

Table 10. Component test results.

4.3. Discussion

The previous works focused on an alphabet-to-alphabet mapping method. Because, how the transliteration is more phonetic than orthographic, without phonetic infomation[10] it

9 with 20 higher rank results.

10 Hangul alphabet has phonetic as well as orthographic. It may be adopted to our method as phoneme. Because one

may be difficult to acquire more relevant result. In the result, 'crepe(keu-le-i-peu/ keu-le-pe) [11]', 'dealer (dil-leo/ di-eol-leo)', 'diode (da-i-o-deu/ di-o-deu)', and 'pheromone (pe-ro-mon/ pe-eo-o-mon)' etc. produce errors in the previous works because they are transliterated into Korean with pronunciation and the patterns can not be acquired from an alphabet-to-alphabet mapping method. For example, 'e' before 'p' in 'crepe' is transliterated into Korean chracters 'e-i' but it is usually transliterated into 'e' in training data.

Origin of English word also contributes performance improvement. For example, words such as 'hittite (hi-ta-i-teu /ha-i-ta-i-teu)', 'hernia (he-leu-ni-a/ heo-ni-a)', 'cafeteria (ka-pe-te-li-a/ ka-pi-te-ri-a)'. In summary, E-K transliteration is not an alphabet-to-alphabet mapping problem but a problem that can be solved with mixed use of alphabet, phoneme, and word formation information.

In the experiments, we find that vowel transliteration is the main reason of errors rather than consonant transliteration in E-K transliteration. Especially, 'AH' is the most ambiguous phoneme because it can be several Korean characters such as 'eo', 'e', 'u', and so on. To improve performance of E-K transliteration, more specific rules may be necessary to handle vowel transliteration.

5. Conclusion

We propose an English-Korean transliteration model using pronunciation and contextual rules. Unlike the previous works, our method use phonetic and orthographic information for transliteration. With them our method showed significant performance increase about 31%. We also showed that origin of English words was important in E-K transliteration.

In future works, a study is attempting to develop a method for handling English of various foreign origin, which this paper did not handle. To improve accuracy, contextual rules must be added using larger data. Our method may be useful to many NLP applications such as automatic bi-lingual dictionary construction, information retrieval, machine translation, speech recognition and so on.

EPU may produce many phonemes, it may be difficult to acquire a good result without context of phoneme and EPU.

[11] English word (correct transliteration / transliteration by the previous works)

References

Brown, P. F. and et al. (1990), "A Statistical Approach to Machine Translation," Computational Linguistics, Vol 16 (2), June.

Covington, M. A., (1996). "An algorithm to align words for historical comparison", Computational Linguistics, 22.

Hall, P., and G. Dowling, (1980), "Approximate string matching," Computing Surveys, 12(4), 381-402.

John Hough, (1953) "Scientific Terminology" New York: Rhinehart & Company, Inc.

Kang, Y. and A. A. Maciejewski, (1996). "An algorithm for Generating a Dictionary of Japanese Scientific Terms", Literary and Linguistic Computing, 11(2).

Kang B.J. and K-S. Choi (2000), "Automatic Transliteration and Back-transliteration by Decision Tree Learning", In Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece.

Kang B.J. and Key-Sun Choi, (2001) "Two approaches for the resolution of word mismatch problem caused by English words and foreign words in Korean information retrieval", International journal of computer processing of oriental language vol 14/No 2, 109-131

Kim J.J., J.S. Lee, and K-S. Choi., (1999). "Pronunciation unit based automatic English-Korean transliteration model using neural network", In Proceedings of Korea Cognitive Science Association (in Korean)

Knight, K. and J. Graehl, (1997). "Machine Transliteration". In Proceedings. of the 35th Annual Meetings of the Association for Computational Linguistics (ACL) Madrid, Spain.

Lee, J. S. and K. S. Choi, 1998. English to Korean Statistical transliteration for information retrieval. Computer Processing of Oriental Languages, 12(1):17-37.

Luschnig, C.A.E. (2001) English word origin, http://www.ets.uidaho.edu/luschnig/EWO

Ministry of culture and tourism, Republic of Korea, "English-to-Korean Standard conversion rule", 1995 (in Korean)

Quinlan,J.R. (1993), "C4.5: Programs for Machine Learning", Morgan Kauffman.