

CHAPTER 3

ENGLISH-TO-KOREAN TRANSLITERATION

3.1 Overview

3.2 Previous Research on English-to-Korean Transliteration

Three types of automatic English-to-Korean transliteration models have been proposed in the literature: grapheme-based models (Lee and Choi, 1998; Jeong, Myaeng, Lee, and Choi, 1999; Kim, Lee, and Choi, 1999; Lee, 1999; Kang and Choi, 2000a; Kang and Kim, 2000; Kang, 2001), phoneme-based models (Lee, 1999; Jung et al., 2000), and ortho-phonemic models (Oh and Choi, 2002, 2005; Oh, Choi, and Isahara, 2006b). Grapheme-based models work by directly transforming source language graphemes into target language graphemes without explicitly utilizing phonology in the bilingual mapping. Phoneme-based models, on the other hand, do not utilize orthographic information in the transliteration process. Phoneme-based models are generally implemented in two steps: first obtaining the source language pronunciation and then converting that representation into the target language graphemes. Ortho-phonemic models consider the joint influence of orthography and phonology on the transliteration process. They also involve a two-step process, but rather than discarding the orthographic information after the pronunciation of a source word has been determined, they utilize it as part of the transliteration process.

3.2.1 Grapheme-Based English-to-Korean Transliteration Models

Grapheme-based transliteration models attempt to define mappings directly from English to Korean orthography.

3.2.1.1 Lee and Choi (1998); Lee (1999)

Lee (Lee and Choi, 1998; Lee, 1999) proposed a Bayesian grapheme-based English-to-Korean transliteration model that generates the most likely transliterated Korean word \hat{K} from an English source word E on the basis of Equation 3.1.

$$(3.1) \hat{K} = \underset{K}{\operatorname{argmax}} P(K|E) = \underset{K}{\operatorname{argmax}} P(E|K)P(K)$$

Lee’s model begins by segmenting an English word into a sequence of graphemes (Deligne et al., 1995; Bisani and Ney, 2002), or multi-letter sequences that correspond to English phonemes. For example, the word *speaking* can be represented as a sequence of five graphemes (from Bisani and Ney, 2002: 105):

$$\begin{array}{cccccc} \textit{speaking} & = & \textit{s} & \textit{p} & \textit{ea} & \textit{k} & \textit{ing} \\ \text{/spikiŋ/} & & \text{/s/} & \text{/p/} & \text{/i/} & \text{/k/} & \text{/iŋ/} \end{array}$$

In order to identify the most likely Korean grapheme for each English grapheme, Lee and Choi (1998) and Lee (1999) generate all possible grapheme sequences for each English word and the corresponding Korean transliteration. For example, the English word *data* can be segmented into the following 8 possible subsequences *data*, *dat-a*, *da-ta*, *da-t-a*, *d-ata*, *d-a-ta*, *d-at-a*, *d-a-t-a*, and the corresponding Korean transliteration *deitə* can be segmented into 16 possible subsequences: *deitə*, *deit-ə*,

dei-t-ə, etc. Maximum likelihood estimates specifying the probability with which each English grapheme maps onto each Korean grapheme are obtained via the expectation maximization algorithm (Dempster, Laird, and Rubin, 1977).

The probability of a particular Korean grapheme sequence $K = (k_1, \dots, k_L)$ occurring is represented as a first-order Markov process (Manning and Schütze, 1999: Ch. 9) and is estimated as the product of the probabilities of each grapheme k_i (Equation 3.2):

$$(3.2) \quad P(K) \cong P(k_1) \prod_{i=2}^L p(k_i | k_{i-1})$$

The probability of observing an English grapheme sequence $E = (e_1, \dots, e_L)$ given a Korean sequence K is estimated from the observed grapheme alignment probabilities as

$$(3.3) \quad P(E|K) \cong \prod_{i=1}^n p(e_i | k_i)$$

This approach suffers from two drawbacks (Oh, Choi, and Isahara, 2006a; Oh et al., 2006b). The first is the enormous time complexity involved in generating all possible grapheme sequences for words in both English and Korean. There are an exponential number of ordered substrings to consider for a string of length L (e.g., string $|L|$ has $2^{|L|-1}$ possible ordered subsequences). Because this number of substrings must be considered for both languages, the approach is impossible to implement for a large number of transliteration pairs. The second consideration involves the nature of the alignment procedure for identifying within-language graphemes. Alignment errors in this stage propagate to the cross-language alignments, leading to incorrect transliterations that might otherwise be avoided. This model obtained recall of 0.47

when evaluating the 20 best transliteration candidates per word in a comparison reported in Jung et al. (2000: 387, Table 3; trained on 90% of an 8368 word data set and tested on 10%). Recall is defined as the number of correctly transliterated words divided by the number of words in the test set.

3.2.1.2 Kang and Choi (2000a,b)

Kang and Choi (2000a, b) describes a grapheme-based transliteration model that uses decision trees to convert an English word into its Korean transliteration. Like Lee and Choi (1998) and Lee (1999), it is based on alignments between source and target language graphemes. However, this approach differs in terms of how the alignments are obtained.

Kang and Choi (2000a, b) explicitly mentions some of the steps undertaken to mitigate the exponential growth of the grapheme mapping problem, noting that the number of combinations can be greatly reduced by disallowing many-to-many mappings and null correspondences from English to Korean. Furthermore, Kang and Choi (2000a, b) does not apply an initial English grapheme-phoneme alignment step, but directly aligns English and Korean graphemes. Character alignments are automatically obtained using a modified version of a depth-first search alignment algorithm based on Covington (1996).

Covington (1996)'s alignment procedure is a variant of the string edit-distance algorithm (Levenshtein, 1966) that treats string alignment as a way of stepping through two words performing a match or skip operation at each step. Kang and Choi (2000a, b) extends Covington's algorithm by adding a bind operation that removes null mappings in the alignment and allows many-to-many correspondences between source and target characters. For example, Covington's edit distance algorithm aligns *board* and /poti/ as

b o a r d -
 p o - - t i

which produces null mappings (the ‘-’ symbol) in both the source and target strings.

Kang and Choi’s modifications produce the following alignment

b oar d
 p o ti

in which the null mapping has been replaced by a binding operation that produces many-to-many correspondences. Kang and Choi further modify the original alignment procedure by assigning different costs to matching symbols on the basis of their phonetic similarity (i.e., phonetically dis-similar alignments such as consonant-vowel receive higher penalties than an alignment between phonetically similar consonants such as /f/ and /p^h/). The penalties are heuristic in nature and are based on the following two observations:

- English consonants tend to transliterate as Korean consonants, and English vowels tend to transliterate as Korean vowels;
- there are typical Korean transliterations of most English characters.

These heuristics are implemented in terms of penalties involving the matching, skipping, or binding of specific classes of English and Korean characters (Kang and Choi, 2000a: 1139, Table 2).

Kang and Choi (2000a, b) models the transliteration process in terms of a bank of decision trees that decide, for each English letter, the most likely Korean transliteration on the basis of seven contextual English graphemes (the left three, the target, and the right three). For example, given the word *board* and its Korean transliteration <potu>, 5 decision trees would attempt to predict the Korean output on the basis of the representations in Table 3.1.

Kang and Choi (2000a, b) used ID3 (Quinlan, 1986), a decision tree learning algorithm that splits attributes with the highest information gain first. Information

>	>	>	(b)	o	a	r	→	p
>	>	b	(o)	a	r	d	→	o
>	b	o	(a)	r	d	>	→	-
b	o	a	(r)	d	>	>	→	-
o	a	r	(d)	>	>	>	→	tu

Table 3.1: Feature representation for transliteration decision trees used in Kang and Choi (2000a, b)

gain is defined as the difference between how much information is needed to make a correct decision before splitting versus how much information is needed after splitting. In turn, this is calculated on the differences in entropies of the original data set and the weighted sum of entropies of the subdivided data sets (Dunham, 2003: 97–98). Kang and Choi (2000b) reports word-level transliteration accuracy of 51.3% on a 7000 item data set (90% training, 10% testing) when generating a single transliteration candidate per English word. Word accuracy is defined as the number of correct transliterations divided by the number of generated transliterations.

3.2.1.3 Kang and Kim (2000)

Kang and Kim (2000) models English-to-Korean transliteration with a weighted finite state transducer that returns the best path search through all possible combinations of English and Korean graphones. Like Kang and Choi (2000a, b), Kang and Kim (2000) employs an initial heuristic-based bilingual alignment procedure. As with Lee and Choi (1998) and Lee (1999), all possible English-Korean graphone chunks are generated from these alignments. Evidence for a particular English sequence transliterating as a particular Korean sequence is quantified by assigning a frequency-based weight to each graphone pair. This weight is computed in terms of a *context*

and an *output*, where context refers to an English graphone e_i and output refers to an aligned Korean graphone k_i as in Equation 3.4 (Kang and Kim, 2000: 420, Equation 4),

$$\begin{aligned}
 \text{weight}(\text{context} : \text{output}) &= \frac{C(\text{output})}{C(\text{context})} \text{len}(\text{context}) \\
 (3.4) \qquad \qquad \qquad &= \text{weight}(e_i : k_i) = \frac{C(k_i \cap e_i)}{C(e_i)} \text{len}(e_i)
 \end{aligned}$$

where $C(x)$ refers to the number of times x occurred in the training set. The weight is multiplied by the length of the English graphone sequence so that longer chunks receive more weight than shorter chunks.

A transliteration network is constructed as a finite state transducer where arcs between nodes are weighted with the weights obtained from the aligned training data. The best transliteration is found via the Viterbi algorithm (Forney, 1973) as the optimal path through the network.

3.2.2 Phoneme-Based English-to-Korean Transliteration Models

Phoneme-based transliteration models map directly from English phonemes to Korean graphemes.

3.2.2.1 Lee (1999); Kang (2001)

Oh et al. (2006a, b) summarizes two phoneme-based transliteration model originally proposed by Lee (1999) and Kang (2001). Lee (1999)'s model generates Korean transliterations from English words through a two-step process. The first step involves the statistical segmentation of English words into graphones using the alignment procedure described in Section 3.2.1.1. At this point, instead of taking the

orthographic component as the representation of an English word, the phonological representation is used instead.

English phonemes are transformed into Korean graphemes on the basis of a set of standard English-to-Korean conversion rules (Korean Ministry of Culture and Tourism, 1995). These rules are expressed as context-sensitive rewrite rules of the form $A_E X_E B_E \rightarrow Y_K$, meaning that the English phoneme X becomes Korean grapheme Y in the context of English phonemes A and B . For example, the following rule

$$j \rightarrow \underset{si}{\underset{si}{<}} / -\#$$

states that English j becomes $<si>$ at the end of words.

This approach suffered from two main problems: the propagation of errors that result from the statistical alignment procedure, and limitations in the set of phonological rewrite rules. Because the standard conversion rules are expressed in terms of phonological natural classes, there is a poor contextual mapping onto the statistically derived phoneme chunks. Furthermore, a great deal of the variability associated with loanword adaptation is simply not amenable to description by contextual rewrite rules.

Kang (2001)'s model takes the pronunciation of English words directly from a pronouncing dictionary without relying on an automatic English grapheme-to-phoneme alignment procedure. Decision trees are constructed which convert English phonemes into Korean graphemes using the training procedure described in Section 3.2.1.2. The only difference between this model and the grapheme-based model described earlier is that the phoneme-based model applies to a phonological representation rather than an orthographic one. A drawback of the model is that it does

not provide a method for estimating the pronunciation of English words not in the dictionary, making it impossible to generalize to a larger set of transliteration pairs.

3.2.2.2 Jung, Hong, and Paek (2000)

Jung et al. (2000) presents a phoneme-based approach to English-to-Korean transliteration that models the process with an extended Markov window consisting of the current English phoneme, the preceding and following English phoneme, and the current and preceding Korean grapheme. The first step of the transliteration process involves converting an English word to a pronunciation string using a pronouncing dictionary. A transcription automaton is used to generate pronunciations for words not contained in the dictionary. The next step involves constructing a phonological mapping table that links English and Korean pronunciation units. Pronunciation units may consist of vowel or consonant singletons, or larger units made up of combinations of consonant and vowel sequences. Mappings are based on hand-crafted rules that come from examining a set of English-Korean transliteration pairs. For each English pronunciation unit, a list of possible Korean transliterations is determined. Some examples are shown in Table 3.2 (Jung et al., 2000: 388–389, Tables 6-1 and 6-2).

<u>English pronunciation unit</u>	<u>Korean orthographic unit(s)</u>
/p/	ㅍ, ㅂ, ㅍ, ㅂ, ㅍ, ㅂ 'p,b,pi,bb'
/s/	ㅅ, ㅆ, ㅅ, ㅆ, ㅅ, ㅆ 's,si,j,ss,jj'
/ur/	ㅓ, ㅜ, ㅜ 'uə,wə'

Table 3.2: Example English-Korean transliteration units from (Jung et al., 2000: 388–389, Tables 6-1 and 6-2)

English pronunciations are aligned with Korean orthographic strings in a two step heuristic-based process. In the first stage, English and Korean consonants are

aligned. The second pass aligns vowels with vowels while respecting the previously determined consonant alignments. Relying on the table of phonological mappings to constrain the alignment procedure results in a unique alignment for each English-Korean pair. For the generation stage of the transliteration process, all possible segmentations of the English word are produced and the segmentation leading to the most likely Korean transliteration is selected as the transliterated output.

Jung et al. (2000) model the transliteration process in terms of the joint probability of an English word and its Korean transliteration, $P(E, K)$. This probability is approximated by substituting the English word E with its segmented phonemic representation S . The joint probability of E and S can be expressed in terms of a conditional probability according to Equation 3.5 (Jung et al., 2000: 385, Equation 2),

$$(3.5) \quad \begin{aligned} \hat{K} &= \underset{K}{\operatorname{argmax}} P(E, K) \\ &\cong \underset{K}{\operatorname{argmax}} P(S, K) = \underset{K}{\operatorname{argmax}} P(K|S)P(S) \end{aligned}$$

where $S = (s_1, s_2, \dots, s_n)$ and $K = (k_1, k_2, \dots, k_n)$, with s_i an English pronunciation unit and k_i a Korean orthographic segment.

In order to determine k_i , four contextual variables are taken into account: the current English segment s_i , the preceding and following English segments s_{i-1} and s_{i+1} , and the preceding Korean segment k_{i-1} . The transliteration term $P(K|S)$ can be approximated as a product of the probabilities of each k_i conditioned on the contextual variables:

$$(3.6) \quad P(K|S) \cong \prod_{i=1}^n P(k_i | k_{i-1}, s_i, s_{i-1}, s_{i+1})$$

The probability of a given English phonemic segmentation S is estimated from a bigram language model:

$$(3.7) \quad P(S) \cong \prod_{i=1}^n P(s_i | s_{i-1})$$

Jung et al. (2000) describes further enhancements to the basic model in terms of estimating backoffs to combat data sparsity and redundancies in feature prediction. In comparing their model to the grapheme-based approach, the authors note that grapheme-based models may have an advantage in transliterating proper names, which are often absent from pronouncing dictionaries (Jung et al., 2000: 388). This model obtains word level transliteration accuracy, defined as the number of correct transliterations divided by the number of generated transliterations of 53% on a data set containing 8368 items (90% training, 10% testing).

3.2.3 Ortho-phonemic English-to-Korean Transliteration Models

More recent research has explored models that combine orthographic and phonemic information in the transliteration process. In general, models that incorporate orthographic and phonemic information outperform models that include only one source of conditioning information.

3.2.3.1 Oh and Choi (2002)

Oh and Choi (2002) considered the joint influence of English orthography and pronunciation on the transliteration process in the form of ortho-phonemic transliteration rules. Oh and Choi’s model begins by applying the heuristic bilingual alignment procedure described in Kang and Choi (2000a, b). English phonological representations are taken from the Carnegie Mellon Pronouncing Dictionary (CMUDICT) (Weide,

1998). English phonemes are converted into Korean graphemes using the Korean Ministry of Culture and Tourism’s standard English-to-Korean conversion rules described in Section 3.2.2.1 (Lee, 1999). Before converting phones, however, an additional layer of linguistic processing is applied to attempt to improve transliteration accuracy. The first step involves an analysis of out-of-dictionary words to see if they can be analyzed as a compound, while the second involves morphological pattern-matching to see if a word can be classified as etymologically Greek.

If a word is not contained in CMUDICT, it is checked to see whether it can be segmented into two substrings that are contained in the dictionary. The segmentation procedure is a left-to-right scan that incrementally splits a word into two at the current index. For example, *cutline* can be segmented into *c+utline*, *cu+tline*, *cut+line*, at which point the pronunciation of both *cut* and *line* are retrieved from the dictionary. In case a pronunciation can not be found after all segmentations have been attempted, one is automatically generated using a decision tree learning algorithm (Quinlan, 1993).

Oh and Choi (2002) observe that English words of Greek origin are often transliterated into Korean exclusively on the basis of orthography. For example, *hernia* /hɜːniə/ is transliterated as 헤루니아 *heylwunia* and *acacia* /əkeɪʃə/ is transliterated as 아카시아 *akhasia*. Oh and Choi (2002) apply prefix and suffix pattern matching to try to identify a word as etymologically Greek. The prefixes and suffixes they use for classifying words as etymologically Greek are shown in Table 3.3 (Oh and Choi, 2002: Table). For these words, a separate grapheme-based transliteration model is employed. For words not classified as Greek, a system of orthographic/phonemic context sensitive rewrite rules is used.

Oh and Choi (2002)’s phoneme-based transliteration model is based on the set of standard English-to-Korean conversion rules described in Section 3.2.2.1. They

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

Table 3.3: Greek affixes considered in Oh and Choi (2002) to classify English loanwords

applied these rules to 200 randomly selected words from CMUDICT and observed transliteration errors in the output. On the basis of these observations, they selected 27 high frequency rules and augmented them with orthographic information. Table 3.4 contains examples of some of these rules (Oh and Choi, 2002: Table).

Orthography	Pronunciation	Transliteration	Examples
C+le	əl	ㄹ ul	assemble bustle eseympul pesul 어셈블 버슬
sm#	zm	즘 cum	barbarism chauvinism papelicum syopinicum 바버리즘 쇼비니즘
or#	ə	ㅓ e	alligator doctor ayllikeyithe tokthe 앨리게이터 독터

Table 3.4: Example transliteration rules considered in Oh and Choi (2002)

An analysis of their results shows that joint orthographic-phonemic rules outperform either grapheme-only or phoneme-only models (word level transliteration accuracy of 56% versus 35% for a grapheme-only model and 41% for a phoneme-only model). One of the biggest sources of transliteration error occurs for words whose English pronunciation must be automatically generated; i.e., out-of-dictionary items (word level transliteration accuracy of 68% when the pronunciation of the source word is known versus 52% when the pronunciation is automatically generated).

3.2.3.2 Oh and Choi (2005); Oh, Choi, and Isahara (2006)

Oh and Choi (2005); Oh et al. (2006b) presents a generalized framework for combining orthographic and phonemic information into the transliteration process. Oh and Choi (2005) applies three different machine learning methods (maximum entropy modeling, decision tree learning, and memory-based learning) to the transliteration task and evaluates the results.

Oh and Choi’s method begins with establishing alignments between English graphemes and phonemes, and then alignments from English grapheme-phoneme pairs to Korean graphemes. English phonological representations are taken from CMU-DICT (Weide, 1998). Alignments are obtained automatically using a heuristically weighted version of the edit distance algorithm (Levenshtein, 1966). The cost schemes are borrowed from Kang and Choi (2000a, b). The first step involves aligning English graphemes with English phonemes ($G_E \rightarrow P_E$) and then aligning English phonemes with Korean graphemes ($P_E \rightarrow G_K$). Using the English phoneme as a pivot, English graphemes are aligned with Korean graphemes ($G_E \rightarrow P_E \rightarrow G_K$). The ($G_E \rightarrow P_E$) alignments are used to construct training data for a procedure that can be used to generate the pronunciation of words that are not in CMUDICT (the actual procedure is not specified).

Oh and Choi model the transliteration process in terms of a function that maps a set of source language contextual features onto a target language grapheme. Four types of features are used: graphemes, phonemes, generalized graphemes, and generalized phonemes. These features are described in Table 3.5 (Oh and Choi, 2005: 1743, Table 6).

Figure 3.1 (Oh and Choi, 2005: 1744, Figure 6) illustrates the principle of using these features to predict the transliteration of the word *board* (보드 ‘bo-di’).

Feature	Possible Values
English Graphemes	$\{a, b, c, \dots, x, y, z\}$
English Phonemes	$\{/AA/, /AE/, \dots\}$
Generalized Graphemes	Consonant (C), Vowel (V)
Generalized Phonemes	Consonant (C), Vowel (V), Semi-vowel (SV), Silence (\emptyset)

Table 3.5: Feature sets used in Oh and Choi (2005) for transliterating English loan-words in Korean

The grapheme currently being transliterated is represented in the center of a context of three preceding and three following features. It can be described in terms of a 28-feature vector consisting of the current grapheme plus six contextual graphemes, the current phoneme plus six contextual phonemes, the current generalized grapheme plus six generalized graphemes, and the current generalized phoneme plus six generalized phonemes.

$$\left\{ \begin{array}{l} G = (\emptyset \ \emptyset \ \emptyset \ \nabla \ o \ a \ r) \\ P = (\emptyset \ \emptyset \ \emptyset \ /b/ \ /o/ \ \emptyset \ /r/) \\ GG = (\emptyset \ \emptyset \ \emptyset \ C \ V \ V \ C) \\ GP = (\emptyset \ \emptyset \ \emptyset \ C \ V \ \emptyset \ C) \end{array} \right\} \rightarrow \text{ㅂ 'b'}$$

Figure 3.1: Feature representation of English graphemes

Oh and Choi apply three machine learning models to the feature representation described in Figure 3.1: maximum entropy modeling, decision tree learning, and memory based learning. The maximum entropy model (Jaynes, 1991; Berger, Pietra, and Pietra, 1996) is a probabilistic framework for integrating information sources. It

is based on the constraint that the expected value of each feature in the final maximum entropy model must equal the expectation of that same feature in the training set. Training the model consists of finding the probability distribution subject to the constraints that has the maximum entropy distribution (Manning and Schütze, 1999: Chapter 16, 589–591). For the decision tree, Oh and Choi used C4.5 (Quinlan, 1993), a variant of the ID3 model described in Section 3.2.1.2 (Kang and Choi, 2000a, b). Memory-based learning is a k -nearest neighbors classifier (Hastie, Tibshirani, and Friedman, 2001). Training instances are stored in memory, and a similarity metric is used to compare a new instance with items in memory. The k most similar items are stored, and the majority class label is assigned to the new instance. Oh and Choi used TiMBL (Tilburg Memory-Based Learner) (Daelemans, Zavrel, van der Sloot, and van den Bosch, 2003), an efficient knn implementation geared towards NLP applications. The results of these comparisons are shown in Table 3.6.

3.2.4 Summary of Previous Research

Table 3.6 contains a summary of the results of previous English-to-Korean transliteration experiments. The reported results are for 1-best transliteration accuracy, defined as the number of correct transliterations divided by the number of generated transliterations, and include a mixture of words whose English pronunciation was automatically generated and words whose English pronunciation was found by dictionary lookup. Because not all results are reported over the same data set using the same methodology, they should be interpreted as representative of the various approaches to English-Korean transliteration rather than as strict comparisons. In general, the combined models outperform models that only include one source of information in the transliteration process. On average, the grapheme-based models are

more accurate than the phoneme-based models, indicating that orthography alone is a more reliable indicator of the form of a transliterated word than phonology alone.

Model	Method	Accuracy
Ortho-phonemic	Max-Ent <small>Oh et al. (2006a: 137, Table 11)</small>	73.3
	TiMBL <small>Oh et al. (2006b: 200, Table VI)</small>	66.9
	Rewrite Rules <small>Oh and Choi (2002: 6, Table 8)</small>	63.0
	Decision Tree <small>Oh et al. (2006b: 200, Table VI)</small>	62.0
Grapheme-based	Weighted FST <small>Kang and Kim (2000: 422, Table 3)</small>	55.3
	Decision Tree <small>Kang and Choi (2000b: 138, Section 5)</small>	51.3
Phoneme-based	Markov Window <small>Jung et al. (2000: 387, Figure 4)</small>	≈53
	Decision Tree <small>Kang (2001), from Oh et al. (2006b: 200, Table VI)</small>	47.5

Table 3.6: Summary of previous transliteration results

It may or may not be worth attempting to straighten out a mischaracterization of the standard English-to-Korean transliteration rules (Korean Ministry of Culture and Tourism, 1995) that is repeated in one strand of English-to-Korean transliteration research:

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’ (Oh and Choi, 2002: 5) .

Second, the EKSCR does not contain enough rules to generate relevant Korean transliterations since its main focus is on a methods of mapping from one English phoneme to one Korean grapheme without the context

of graphemes and phonemes. For example, the English word *board* and its pronunciation /B AO R D/ are incorrectly transliterated into ‘bo-reu-deu’ by EKSCR. However, the correct one, ‘bo-deu’, can be acquired when their contexts are considered (Oh and Choi, 2005: 1740).

The other problem is that EKSCRs does not contain enough rules to generate relevant Korean transliterations for all the corresponding English words since its main focus is on mapping from one English phoneme to one Korean grapheme without considering the context of graphemes and phonemes. For example, the English word *board* and its pronunciation /B AO R D/ are incorrectly transliterated into “boreudeu” by EKSCRs. If the contexts are considered, they are correctly transliterated into “bodeu” (Oh et al., 2006b: 191).

While it is true that the standard conversion rules do not adequately encapsulate the various ways in which English phonemes transliterate into Korean, the characterization of them as focusing mainly on a one-to-one bilingual mapping in the absence of contextual information is misleading. It is also incongruent with the description of the transliteration rules as “context-sensitive rewrite rules” given in (Oh et al., 2006a: 123). Instead, the rules are expressed in traditional phonological terms of phonologically conditioned sound change.

However, there is no rule that explicitly deals with the conversion of //r/ into Korean in this context. This is because the rules focus on alternations in the pronunciation of English phonemes, i.e., environmentally conditioned changes. /r/ is always dropped in this context, so no rule is included. Nothing predicts that *board* would transliterate as *polutu*. On the other hand, there are lots of examples of

post-vocalic /r/ followed by a consonant that would indicate that board would not transliterate as *polutu* (Korean romanization not part of the original):

1.3 part [pa:t]	파트	<i>phatu</i>
3.2 shark [ʃa:k]	샤크	<i>syakhu</i>
5.1 corn [ko:n]	콘	<i>khon</i>
9.1 word [wə:d]	워드	<i>wetu</i>
9.2 quarter [kwɔ:tə]	쿼터	<i>khwethe</i>
9.3 yard [ja:d]	야드, yearn [yɜ:n]	<i>yatu, yen</i>

So while the general sentiment is true, repeating this same example over and over results in a mischaracterization of the standard conversion rules to the larger research community.

3.3 Experiments on English-to-Korean Transliteration

This section describes and analyzes two ortho-phonemic models for transliterating English loanwords into Korean. The first model is based on a set of phonological conversion rules that describe the changes English words undergo when they are borrowed into Korean. The second model is a statistical model that produces the highest scoring Korean transliteration of an English word based on a set of combined orthographic and phonemic features. The behavior of these two models with respect to the amount of training data required to produce optimal results is examined, and the models are compared to each other in terms of the accuracy of the transliterations each produces. Both models are compared to a maximum entropy transliteration model which has obtained state-of-the-art results in previous research, and scenarios for which each of the models exhibit particular advantages are discussed.

The sections below report the results of a series of experiments on English-to-Korean transliteration. The first experiment deals with the rule based transliteration

model, first describing it in detail and then reporting the results of using it to transliterate a set of English-Korean loanwords. The second experiment presents a modified version of the rule based model which incorporates orthographic information into the transliteration process and examines its transliteration accuracy. The third experiment presents a statistical transliteration model and compares its performance to both the rule based models and a maximum entropy transliteration model. The last section of the chapter summarizes the characteristics of each model with respect to their applicability to situations where aligned bilingual training data is easily obtainable versus situations where it is harder to obtain.

3.3.1 Experiment One

3.3.1.1 Purpose

The purpose of this experiment is to investigate the use of phonological conversion rules for transliterating English words into Korean.

3.3.1.2 Description of the Transliteration Model

The transliteration model used in this experiment is a regular expression-based implementation of the Korean Ministry of Culture and Tourism (1995)'s set of English-to-Korean standard conversion rules. Although prescriptive in tenor, these rules are expressed in terms of feature-based phonological classes and are congruent with descriptive accounts of English loanword adaptation in Korean (e.g., stop and fricative adaptation (Kang, 2003; Kenstowicz, 2005; Lee, 2006; Park, 2007); vowel substitution (Yang, 1996)). The Korean Ministry of Culture and Tourism (1995)'s set of English-to-Korean standard conversion rules were manually converted into regular expressions in a computer program that takes a phonological representation of an English word

as input and produces a Korean transliteration of it as output. The programming language used was Python¹, although any language which provides regular expression support is suitable.

In this experiment, the transliteration process was modeled in three steps. First, a preprocessing step is applied to the English phonological representations that expands the single character representation of diphthongs used by the Hoosier Mental Lexicon (Nusbaum et al., 1984) into two vowel symbols. This step is performed because it reduces the number of symbols and transformation rules needed for transliteration. The second step consists of the successive application of a sequence of regular expression substitutions which transform a string of English phonemes into a Korean phonological representation. Finally, an optional post-processing step may be performed to syllabify the Korean string and convert it to hangul.

This transliteration model assumes the definition of the following two character classes.

```
:shortvowel: = IE@aUcx^  
:vowel: = ieou + :shortvowel:
```

In addition to these definitions, a set of intermediate vowel symbols was used to handle word boundaries and epenthesis and /r/ deletion. # is inserted at the beginning and end of words; ~ serves as a placeholder for deleted /r/, and ! and % stand for the epenthetic vowels /i/ and /i/, respectively. Reserving extra symbols for epenthetic vowels facilitates the application of the phonological conversion rules such that rules that apply later are not inadvertently triggered by a vowel that was not present in the input. The preprocessing step consists of the following six character expansions.

Y -> ai

¹Distributed under an open source license: <http://www.python.org>.

O -> oi
 e -> ei
 W -> au
 X,R -> xr, xr

The transliteration step consists of the following regular expression substitutions, applied in the order presented below. In the description below, the following conventions for representing regular expression substitution are employed. Brackets [] are used to enclose a class of characters; e.g., [:vowel:] stands for any character that is a vowel. ^ inside brackets negates the character class; e.g., [^:vowel:] stands for any character that is not a vowel. Parentheses () are used to enclose regions of the regular expression that can be referred to in the substitution phase by index. Regions are numbered consecutively from the left starting at 1. For example, in the expression (first)(class), \1 refers to first and \2 refers to class. Text starting at %% contains examples meant to illustrate the application of each regular expression, but is not part of the regular expression itself.

1. /r/ deletion

r([[:vowel:]]) -> ~\1 %% e.g., 'church' #CxrC# -> #Cx~C#

2. /ts, dz/ epenthesis

ts([[:vowel:]]) -> C!\1 %% e.g., 'Pittsburgh' #pItsbx~g# -> #pIC!bx~g#

dz([[:vowel:]]) -> J!\1 %% e.g., 'odds', #adz# -> #aJ!

3. voiceless obstruent epenthesis

([[:shortvowel:]])([ptk])([[:vowel:]]) -> \1\2!\3

%% e.g., 'cape' #keip# -> #keip!#

([sTf])([[^]:vowel:]) -> \1!\2

%% e.g., 'first' #fx~st!# -> #fx~s!t!#

4. voiced obstruent/affricate epenthesis

([vbdzg])([[^]:vowel:]) -> \1!\2 %% e.g., 'cape' #keip# -> #keip!#

([CJSZ])([[^]:vowel:]) -> \1%\2 %% e.g., 'church' #Cx~C# -> #Cx~C%#

5. short vowel voiceless stop substitution

([:shortvowel:])p([[^]:vowel:]) -> \1b\2

%% e.g., 'apt' #@pt!# -> #@bt!#

([:shortvowel:])t([[^]:vowel:]) -> \1d\2

([:shortvowel:])k([[^]:vowel:]) -> \1g\2

6. /l/ gemination

([:vowel:~!%])l([:vowel:]) -> \1ll\2

%% e.g., 'clasp' #k!l@s!p!# -> #k!ll@s!p!#

7. unconditioned consonant substitutions

f -> p

v -> b

T -> s

D -> d

[zZ] -> J

8. unconditioned vowel substitutions

c -> o

I -> i

x -> ˆ
[U|] -> u
@ -> E

3.3.1.3 Experimental Setup

This experiment used the list of 10,000 English-Korean loanword pairs described in 2.1. The phonological representation of each English item in the list was transliterated via the rule based model and the resulting form was compared to the actual Korean adaptation of that English source word. Because the rule based model does not require training data, it was applied to all of the items in the data set.

3.3.1.4 Results and Discussion

The first evaluation of the rule based transliteration model measured transliteration accuracy in terms of the number of transliterated items that exactly matched the actual Korean form. Overall transliteration accuracy, measured as

$$\frac{\# \text{ of correct transliterations}}{\# \text{ of actual transliterations}}$$

was 49.2%. A strict comparison between the current work and previous research is not feasible given the range of approaches represented therein on different data sets². However, these results are in line with previous phoneme-based approaches ($\approx 53\%$ reported in Jung et al., 2000; 47.5% reported in Kang, 2001).

Based on the analysis of English loanwords in Korean provided in 2.1, it is known that vowel transliteration is harder to predict by phonological rule than consonant transliteration (Table 2.5). Therefore, we also examined the performance of

²Repeated efforts to obtain access to previously used data sets were unsuccessful.

the rule based model in terms of the number of correctly transliterated consonants per item. This comparison was made by deleting input vowels from both the predicted form (after transliteration) and the actual form, and comparing the remaining sequence of consonants. An input vowel is a transliterated vowel whose presence in the transliterated form is due to a direct mapping from the original English vowel phoneme. In other words, epenthetic vowels were retained in the predicted and actual forms. For example, given the English word *pocket* and actual transliteration of 포켓 *phokheys*, a predicted transliteration of 파켓 *phakheys* counts as containing all correctly transliterated consonants ($phkhs = phkhs$).

Consonant sequence transliteration accuracy, defined as

$$\frac{\# \text{ of correct consonant sequence transliterations}}{\# \text{ of consonant sequence transliterations}}$$

was 89.9%. This is a stricter measure than overall character accuracy (cf. Kang and Kim, 2000; Oh and Choi, 2002), because it requires that all consonants in a word are correctly generated and ordered to count as correct. It also requires that rules concerning vowel epenthesis have correctly applied, as these rules often change the nature of the preceding consonant (e.g., whether it is an aspirated syllable onset or an unaspirated coda).

The congruence of the full word transliteration results with previous models and the disparity between full word transliteration and consonant sequence transliteration reported here suggest that the phonological information represented in this data set alone does not convey sufficient information to reliably predict the transliterated form of vowels in English loanwords in Korean. On the basis of this observation and the analysis of English loanwords in Chapter 2.1, we modified the rule based model to incorporate orthographic information into the transliteration of vowels. This modified rule based transliteration model is described in the next section.

3.3.2 Experiment Two

Previous researchers have examined the performance of transliteration models that produce a set of transliteration candidates for a given input string (Lee, 1999; Jung et al., 2000; Kang and Kim, 2000). The motivation for this approach to transliteration is spelled out in Kang and Choi (2000b), which points out that multiple transliterations of the same English word are often found in large document collections, creating problems for information retrieval. For example, the English word *digital* appears variously in Korean as *ticithel*, *ticithal*, and *ticithul* even though *ticithel* is the standard transliteration (Kang and Choi, 2000b: 133). Following this strand of research, this experiment examines the performance of a rule based model that produces a set of transliteration candidates.

3.3.2.1 Purpose

The purpose of this experiment is to investigate the performance of an ortho-phonemic rule based transliteration model for generating sets of transliteration candidates for English loanwords in Korean.

3.3.2.2 Description of the Model

One of the main sources of transliteration variability for vowels lies in the effect of orthography on pronunciation, where the orthographic vowels ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ are often transliterated with their IPA values of /a,e,i,o,u/ regardless of their actual pronunciation (Oh and Choi, 2005). Therefore, we modified the rule-based model to produce both orthographic and pronunciation-based version of English vowels. This model was modified to accept an aligned orthographic and phonological representation of an English word. For each phonological vowel in the input up to two transliterations are produced: one is based on phonological substitution and the other is based on orthographic copying. In case the phonological value of a vowel is equivalent to its orthographic representation (e.g., *smoke* /smok/) only one vowel transliteration is produced.

Prior to transliteration, the alignment between an orthographic and phonemic representation of a word is converted into a finite state automaton whose arcs are labeled with phonemes. A vowel alignment produces up to two arcs from a preceding to a following state. One arc is labeled with a phoneme symbol and the other is labeled with an orthographic character. An example finite state automaton is shown in Figure 3.2 for the alignment between the orthographic and phonological representations for *cactus-k@ktxs*. We used the AT&T Finite-State Machine Library (Mohri, Pereira, and Riley, 1998) to process the finite state automata produced for transliteration. Taking all paths through the finite state automaton in Figure 3.2 yields four strings which are each input to the rule based transliteration model described in Experiment 1: *k@ktxs*, *k@ktus*, *kaktxs*, *kaktus*. If a phonological vowel aligns with more than one orthographic vowel, e.g., *head:hE-d*, the only orthographic vowel produced is the one aligned directly to the phonological vowel. In other words, the null symbol ‘-’ in

the phonological representation does not produce any additional paths through the finite state machine. In principle, if a word contains V phonological vowels, up to

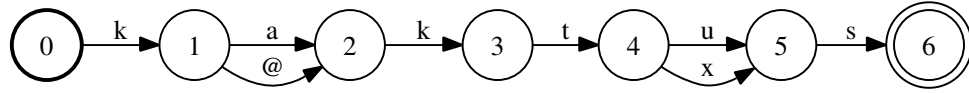


Figure 3.2: Example rule-based transliteration automaton for *cactus*

2^V unique transliterations may be produced: every vowel may result in two paths through the finite state automaton, so the final number of transliteration candidates will be $2v_1 \times 2v_2 \dots \times 2v_V$. In practice, because the orthographic and phonemic vowels are often equivalent, far fewer candidates are produced (average 3.4 per word).

3.3.2.3 Experimental Setup

This experiment used the list of 10,000 English-Korean loanword pairs described in 2.1. The aligned orthographic and phonological representation of each English item in the list was transliterated via the ortho-phonemic rule based model and the resulting forms were compared to the actual Korean adaptation of that English source word. Because the ortho-phonemic rule based model does not require training data, it was applied to all of the items in the data set.

3.3.2.4 Results and Discussion

Following Lee (1999), Jung et al. (2000) and Kang and Kim (2000), we report whole word transliteration accuracy as the average number of correctly transliterated words divided by the actual number of loanwords (recall)

$$\frac{\# \text{ of correct transliterations}}{\# \text{ of actual transliterations}}$$

We also report macroaveraged transliteration precision, which takes into account the total number of transliteration candidates produced

$$\frac{\# \text{ of correct transliterations}}{\# \text{ of generated transliterations}}$$

The ortho-phonemic rule based model returns recall and precision values of 0.78, 0.23. In applications such as bilingual information retrieval where the cost of false positives are low or the chance of generating a false hit is unlikely (Kang and Choi, 2000b, 2002), this model offers benefits over the rule based model in terms of coverage. Once again, these results are compatible with previous research that has reported transliteration accuracy over multiple transliteration candidates (Lee 1999; Jung et al. 2000; Kang and Kim 2000). However, the current model offers two advantages over previous statistical approaches to English-Korean transliteration. One is that a rule based approach does not require a bilingual training set. Its only requirement is a monolingual pronunciation dictionary, which for English at least is readily available (Weide, 1998). This means that a rule based approach to transliteration can be extended to a large number of language pairs more quickly and with less expenditure of resources than approaches that require aligned bilingual data (see Section 3.3.5 for elaboration of this point).

A second advantage of the current model over previous n -best approaches is that by focusing attention on the transliteration units that exhibit the most variability (vowels), we are able to generate a relatively small number of transliteration candidates per word. Furthermore, the set of candidates is tuned to the input in such a way that relatively invariant items (e.g., a word with one phonological vowel whose pronunciation matches its orthographic form like *smoke* /smok/) produce a small set

of transliteration candidates. Inputs that are likely to exhibit greater variation produce larger candidate sets. Finally, we are able to offer a direct comparison between the current approach and previous ones in terms of the precision given a correctly generated transliteration. On average, when the correct transliteration appears in the candidate set the ortho-phonemic rule based model generates 2.85 candidates, giving a precision when correct of $1/2.85 = 0.35$. The size of the candidate set considered by previous researchers varies – Lee (1999) evaluated transliteration accuracy on the basis of the 20 most likely transliteration candidates, giving a precision when correct of 0.05; Jung et al. (2000) considered the top 10 transliteration candidates giving a precision when correct of 0.10, and Kang and Kim (2000) used the top 5, giving a precision when correct of 0.20, all of which are considerably lower than the current results.

Although the relative performance of the ortho-phonemic transliteration model represents an improvement over previous work, its overall precision is quite low. A further disadvantage of the model is that it does not rank transliteration candidates by any measure of goodness. Many statistical models do allow an ordering of a set of transliteration candidates. Therefore, we conducted a third experiment with a statistical transliteration model that produces a ranked list of transliteration candidates, and compare its performance to the rule based models.

3.3.3 Experiment Three

3.3.3.1 Purpose

The purpose of this experiment is to examine the performance of a statistical transliteration model and compare it to the ortho-phonemic rule based model in terms of ranking transliteration candidates.

3.3.3.2 Description of the Model

In this experiment we model the task of producing a transliterated Korean character in terms of the probability of that character being generated by a given sequence of graphemes and phonemes. Under this approach, the task of transliterating an English word into Korean can be formulated as the problem of finding an optimal alignment between three streams of symbols

$$\begin{pmatrix} G_E = g_1, \dots, g_L \\ \Phi_E = \varphi_1, \dots, \varphi_L \\ K = \kappa_1, \dots, \kappa_L \end{pmatrix}$$

where G_E is a sequence of English graphemes, Φ_E is a sequence of English phonemes, and K is a sequence of Korean graphemes. We assume that the three sequences have equal length (L) due to the insertion of a null symbol ('-') when necessary, and assume a one-to-one alignment between symbols in the three strings. For example, the English word 'first' and its Korean transliteration 피스트 /p^hΛsit^hi/ can be represented as

$$\begin{pmatrix} G_E = & f & i & r & s & - & t & - \\ \Phi_E = & f & ɜ & - & s & - & t & - \\ K = & p^h_1 & \Lambda_2 & -_3 & s_4 & i_5 & t^h_6 & i_7 \end{pmatrix}$$

with the symbol alignments (f, f, p^h) , $(i, ɜ, \Lambda)$, $(r, -, -)$, etc.

We are interested in obtaining the Korean string K that receives the highest score given (G_E, Φ_E, K) . Computing the score of (G_E, Φ_E, K) can be formulated as a decoding problem that consists of finding the highest scoring Korean string \hat{K} given the aligned sequences of English graphemes and phonemes G_E and Φ_E .

The score of a particular Korean string given G_E and Φ_E is the product of the scores of the alignments comprising the three sequences:

$$\text{Score}(K|G_E, \Phi_E) = \prod_{i=1}^L p(\kappa_i|g_i, \varphi_i)$$

In order to account for context effects of adjacent graphemes and phonemes on the transliteration of a particular English grapheme-phoneme pair, we define \mathbf{g}_i and φ_i as subsequences of G_E and Φ_E , respectively, centered at i and containing elements $\langle g_{i-2}, \dots, g_{i+2} \rangle$ and $\langle \varphi_{i-2}, \dots, \varphi_{i+2} \rangle$, respectively. For example, if $\kappa_4 = s$ in the preceding example, then $\mathbf{g}_4 = \langle i, r, s, -, t \rangle$ and $\varphi_4 = \langle ʃ, -, s, -, t \rangle$. Positions $i < 1$ and $i > L$ are understood to contain a boundary symbol ($\#$) to allow modeling context at word starts and ends. We estimate the probability of κ_i given subsequences \mathbf{g}_i and φ_i with relative frequency counts:

$$p(\kappa_i|\mathbf{g}_i, \varphi_i) = \frac{p(\mathbf{g}_i, \varphi_i, \kappa_i)}{p(\mathbf{g}_i, \varphi_i)} \approx \frac{c(\mathbf{g}_i, \varphi_i, \kappa_i)}{c(\mathbf{g}_i, \varphi_i)}.$$

Given the relatively large context window (2 preceding and 2 following orthographic phoneme pairs), the chance of encountering an unseen feature in the test set is relatively high. In order to mitigate the effect of data sparsity on the transliteration model described above, we modified it to use a backoff strategy that involved successively decreasing the size of the context window centered at the Korean character currently being predicted until a trained feature was found. The specific backoff strategy used in this model is to search for features in the following order starting at the top of the list, where S_i represents the source orthographic-phoneme pair at the index of the Korean letter being predicted and s_i represent preceding and following ortho-phonemic pairs:

$$\begin{aligned}
& s_{i-2}s_{i-1}S_i s_{i+1}s_{i+2} \\
& s_{i-2}s_{i-1}S_i s_{i+1} \\
& \quad s_{i-1}S_i s_{i+1}s_{i+2} \\
& \quad s_{i-1}S_i s_{i+1} \\
& s_{i-2}s_{i-1}S_i \\
& \quad S_i s_{i+1}s_{i+2} \\
& \quad s_{i-1}S_i \\
& \quad \quad S_i s_{i+1} \\
& \quad \quad S_i
\end{aligned}$$

As soon as a trained feature is found, iteration stops and the most highly ranked Korean target corresponding to that feature is produced. In the event that no feature corresponding to S_i is found, no prediction is made. This backoff strategy was based on the intuition that larger contextual units provide more reliable statistical cues to the transliteration of an English segment; it was determined prior to assessing its performance on any of the data and was not altered in response its performance on the data.

In order to establish a comparison between previous statistical transliteration approaches and the current work, we also applied a maximum entropy model (Berger et al., 1996; Pietra, Pietra, and Lafferty, 1997) that was demonstrated to outperform other machine learning approaches to English-Korean transliteration in previous comparisons (Oh and Choi, 2005; Oh et al., 2006a). The maximum entropy model is a

conditional probability model that incorporates a heterogeneous set of features to construct a statistical model that represents an empirical data distribution as closely as possible (Berger et al., 1996; Zhang, 2004). In the maximum entropy model, events are represented by a bundle of binary feature functions that map an outcome even y and a context x to $\{0, 1\}$. For example, the event of observing the Korean letter ‘ p ’ in the context of $\#\#boa$ in a word like *board* can be represented as

$$f(x, y) = \begin{cases} 1 & \text{if } y=p \text{ and } x=\#\#boa \\ 0 & \text{otherwise.} \end{cases}$$

Once a set of features has been selected, the corresponding maximum entropy model can be constructed by adding features as constraints to the model and adjusting their weights. The model must satisfy the constraint that the empirical expectation of each feature in the training data equals the expectation of that feature with respect to the model distribution. Among the models that meet this constraint is one with maximum entropy. Generally, this maximum entropy model is represented as

$$p(y|x) = \frac{1}{Z(x)} \exp \left[\sum_{i=1}^k \lambda_i f_i(x, y) \right]$$

where $p(y|x)$ denotes the conditional probability of outcome y given contextual feature x , k is the number of features, $f_i(x, y)$ are feature functions, and λ_i is a weighting parameter for each feature. $Z(x)$ is a normalization factor defined as

$$Z(x) = \sum_y \exp \left[\sum \lambda_i f_i(x, y) \right]$$

to guarantee that $\sum_y p(y|x) = 1$ (Berger et al., 1996; Zhang, 2004).

In this experiment, we used Zhang Le’s maximum entropy toolkit (Zhang, 2004). In addition to the contextual features used by the statistical decision list model proposed here, we added grapheme-only and phoneme-only contextual features to the maximum entropy model in order to provide a close replication of the feature sets described by Oh et al. (2006a, b). Thus, each target character k_i is represented by a bundle of orthographic, phonemic, and ortho-phonemic contextual features. The full feature set is represented in Table 3.7 for the transliteration of target ‘ p ’ in the word *board*.

Feature		Target
Orthographic	##boa, ##bo, #boa, #bo, ##b, boa, #b, bo, b	‘ p ’
Phonemic	##bo-, ##bo, #bo-, #bo, ##b, bo-, #b, bo, b	‘ p ’
Ortho-phonemic	##boa:##bo-, ##bo:##bo, #boa:#bo-, #bo:#bo, ##b:##b, boa:bo-, #b:#b, bo:bo, b:b	‘ p ’

Table 3.7: Feature bundles for transliteration of target character ‘ p ’

3.3.3.3 Experimental Setup

We evaluated both models by splitting the list of loanwords used in Experiments 1 and 2 into a training set and a disjoint set used for testing. 10% of the data was fixed as a test set, and the remainder of the total data set was used to select training data. The size of the training split ranged from 5% (500 items) to 90% (9000 items) of the total

data set in 5% intervals. Each training split was tested on the same 10% test set. This procedure was repeated 10 times, and the results were averaged. Following Oh and Choi (2005), we trained the maximum entropy model using the default Gaussian prior of 0; in addition we used 30 iterations of the default L-BFGS method of parameter estimation and did not change any other default settings. However, we note that a training regime which utilizes development data to tune the Gaussian parameter and uses more training iterations may produce better results than those obtained here.

3.3.3.4 Results and Discussion

The first evaluation of the statistical transliteration models is reported in terms of 1-best whole word transliteration accuracy, defined as

$$\frac{\# \text{ of correct transliterations}}{\# \text{ of actual transliterations}}.$$

Figure 3.3 depicts transliteration accuracy for the two statistical models as a function of size of the training set. This figure also shows the performance of the rule based model for comparison. Because the rule based model does not require training data, its performance is flat. For both statistical models, transliteration accuracy clearly depends on the amount of training data. As the amount of training data increases, the performance of the maximum entropy model and the statistical model proposed here nearly converge, but for all trials reported here the performance of the maximum entropy model never exceeds the performance of the newly proposed model.

The best transliteration accuracy obtained by the statistical transliteration model and the maximum entropy models is 73.4% and 71.9%, respectively. The proposed model is relatively robust even to small amounts of training data, performing better than the rule based model with as few as 500 training items (5% training data).

The performance difference between the statistical decision model and the maximum entropy model is most noticeable for small amounts of training data. On 500 training items (5% training data), the statistical decision model performs nearly 20 percentage points higher than the maximum entropy model, indicating a potential advantage for the use of this model in situations where training data is scarce.

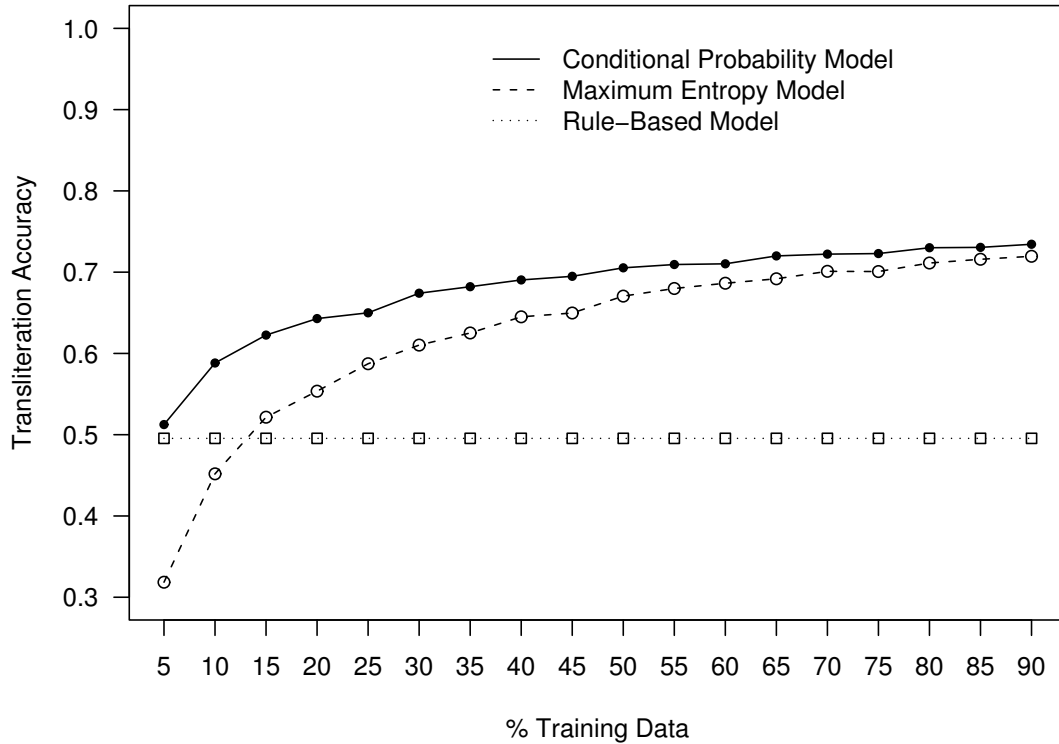


Figure 3.3: Performance of three transliteration models as a function of training data size

We also examined the performance of the statistical decision model with respect to transliteration accuracy of consonant sequences (Experiment 1). The statistical decision model returns 90.8% consonant sequence transliteration accuracy, comparable to that of the rule based model (89.9%). These facts suggest that consonant transliteration is decidedly less variable than vowel transliteration, and that the

main advantage that statistical models have over the rule based model is in accounting for the contextual effects of orthography on vowel transliteration.

In order to compare the statistical decision model to the ortho-phonemic rule based model under the condition of producing multiple transliteration candidates, the statistical model was modified to produce up to two Korean characters for each phonological vowel in an English input. For example, given an input of *cactus-k@ktxs*, the model produces a weighted finite state automaton whose weights correspond to the negative log probabilities of each Korean character given a source feature (Figure 3.4). Transliteration candidates are ranked according to the cost of their path through

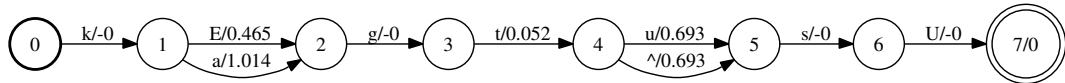


Figure 3.4: Example probabilistic transliteration automaton for *cactus*

the finite state automaton. For the *cactus* example, we obtain the following ranking of transliteration candidates: *kEgtusU*, *kEgt^sU*, *kagtusU*, *kagt^sU*, with the correct transliteration *kEgt^sU* coming in second place.

Figure 3.5 contains precision and recall curves as a function of the amount of training data for the statistical decision list model producing multiple transliteration candidates. When trained on 90% of the data, the statistical model obtains recall and precision scores of 0.84 and 0.49. The rule based model returns recall and precision values of 0.78 and 0.23, and does not systematically vary with respect to the amount of training data. One reason for the higher precision of the statistical model is that it generates on average fewer candidates than the rule based model – 1.9 versus 3.4, respectively. The reason that the statistical model generates fewer candidates is that very often the second Korean character produced by the decision list is the same as the first, in which case the model only makes one prediction.

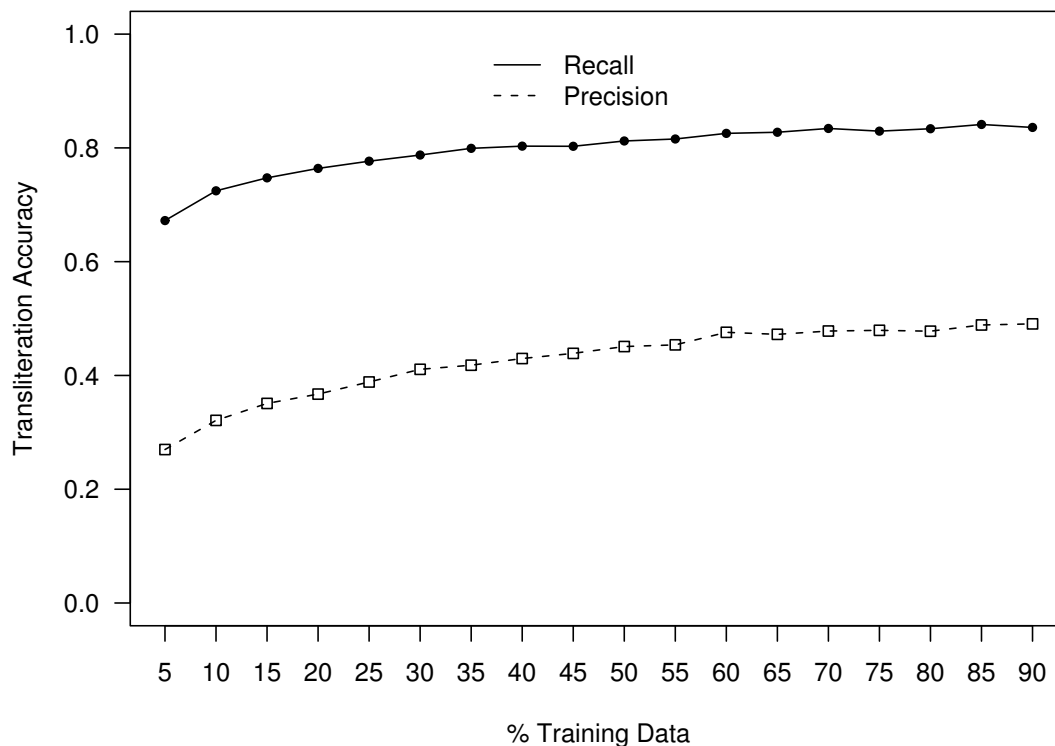


Figure 3.5: Performance of the statistical decision list model producing multiple transliteration candidates as a function of training data size

This situation occurs when a more specific feature predicts a single vowel, and in order to obtain the second transliteration candidate, the backoff model described above is traversed, and the next feature encountered also predicts the same vowel. When this happens only one transition for that feature is generated in the corresponding finite state automaton. In this way the statistical decision list model is taking advantage of converging statistical evidence to limit the number of candidates it produces.

The statistical decision list model also offers an advantage over the ortho-phonemic rule based model in that it is capable of producing a ranked list of transliteration candidates, with the best candidate appearing at the beginning of the list. In order to compare the statistical model with the rule based model in terms of candidate ranks, we computed the mean reciprocal rank. The reciprocal rank of a transliteration candidate is the multiplicative inverse of the rank of that candidate. For example, if the correct transliteration occurred as the second candidate in the list, that item’s reciprocal rank is $1/2 = 0.5$. The mean reciprocal rank is the average of the reciprocal ranks of each transliterated item. In case the correct answer does not appear in the list of transliteration candidates for a given item, a reciprocal rank of 0 is assigned. The mean reciprocal rank for the statistical model is 0.77 versus 0.54 for the rule based model³.

3.3.4 Error Analysis

Examination of transliterations missed by the statistical model shows that many of these items are ones for which vowel transliteration follows an orthographic transliteration, e.g., *oxalis* → /oksallisi/, *orangutan* → /olaju^han/, *ketene* → /k^heten/, *antivitamin* → /ant^hipit^hamin/, *delphi* → /tel^hi/, *lazuli* → /lat^hulli/, *alkali* → /alk^halli/. An alternative explanation for orthographic transliteration is that the word is not borrowed directly from English but is borrowed in both languages from another source or has come to Korean from English via Japanese (Kang, Kenstowicz, and Ito, 2007). Although the ability to assess a detailed etymological history of newly encountered foreign words is difficult to implement in an automatic transliteration system, knowledge of the frequency of a word’s usage in non-English text (such as

³The rule based model does not impose a ranking on transliteration candidates, so the default hash order of the Python dictionary object was used to order candidates in the rule based model.

would be available, e.g., from Google estimates of language specific document counts for a word) could be explored for its utility in influencing the expectation of an English phonological versus orthographic transliteration. Work along these lines remains for future research.

A second area where both the statistical and rule-based models had difficulty is consonant transliteration corresponding to internal word boundaries in compounds like *taphole*, *spillover*, *blackout*, *kickout*, *locknut*, and *cakework*. In these cases the actual transliterations mark the presence of the internal word boundary by applying the expected end of word transliteration rule. For example, in the transliteration of the word *black*, the final /k/ becomes an unaspirated coda in Korean: /pʰɪllæk/. In intervocalic position, English voiceless stops typically aspirate and are realized as syllable onsets. For example in the word *Utah*, the English /t/ becomes Korean /tʰ/, as in /yutʰa/. In compound words like *blackout*, however, the intervocalic stop follows the end of word transliteration pattern and becomes /pʰɪllækʰaus/. This transliteration is unexpected if only the segmental context is considered, where the intervocalic consonant would typically become an onset of the following syllable *black-out* → */pʰɪllækʰaus/). Applying a module to pre-identify potential compound words and insert a word boundary symbol (e.g., *blackout* → #black#out#) is one way to incorporate additional morphological knowledge into the transliteration process and would be expected to improve transliteration accuracy in these cases.

3.3.5 Conclusion

This chapter presented two novel transliteration models, both of which are robust to small amounts of data and are parsimonious in terms of the number of parameters required to estimate them and the number of outputs they produce. The rule

based model is defined by a small set of regular expressions and requires no training data. By modifying it to produce both orthographic and pronunciation based vowel transliterations, its coverage is substantially increased. Relative to previous n -best transliteration models, its precision is high; however, its precision is substantially lower than that of the statistical decision list model when the latter model is modified to produce multiple transliteration candidates as well.

The statistical decision list model achieves reasonable results on small amounts of training data. As the amount of training data increases, the performance of the two statistical models becomes much closer, although the simpler statistical model slightly outperforms the maximum entropy model on all trials in the experiments reported here. However, the maximum entropy model provides greater flexibility for incorporating multiple sources of information, and its performance may increase given a richer feature set for which the statistical decision list model is less suited. Furthermore, its performance may improve given a suitable Gaussian penalty. These possibilities remain to be explored in future research.

The rule based and statistical models lend themselves to situations where bilingual training data is scarce or unavailable. Although the cost of developing an aligned list of loanwords for an arbitrary pair of languages may be lower than the cost of developing a richer lexical resource such as a large syntactically and semantically annotated corpus, it is not negligible. We are not aware of any accounts of the cost of developing a list of aligned English-Korean loanwords from scratch, but can provide an estimate of the amount of data that would be required to produce a similar list of English loanwords in Chinese.

Chinese is similar to Korean in that it has recently begun importing English loanwords into its lexicon as well (Riha and Baker, 2008a, b). However, in Chinese,

these words are often borrowed “as is”, i.e., in the original English orthography. Because these words occupy a distinct range of character codes when stored in electronic orthographic form, they are easy to extract from Chinese text using standard regular expression utilities (e.g., Perl or grep). Figure 3.6 displays the number of unique Roman letter strings in the 2004 CNA subsection⁴ of the Chinese gigaword corpus (Graff, 2007) against the number of Chinese characters read before encountering each new instance. For example, the figure shows that in order to come across 5,000 unique Roman letter words, 17 million Chinese characters have to be read (conservatively, 4.25 million words on the basis of estimates average length of Chinese words in Teahan, Wen, Mcnab, and Witten 2000); in order to extract 10,000 unique Roman letter words, 37 million Chinese characters (9.25 million words) have to be read.

For language pairs that are not as well attested (e.g., Danish-Korean, Italian-Korean), the amount of material required to produce similar lists would be substantially greater or non-existent at the requisite scale. However, phonological accounts of loanword adaptation such as that provided by Li (2005) contain phonological conversion rules for adapting loanwords into Korean from many languages, including Danish, Italian, Thai, Romanian, and Swedish among others. Furthermore, it is possible to find similar accounts for additional pairs of languages like French and Vietnamese (Barker, 1969). In such situations, the cost and time required to develop even a moderately sized list of aligned loanwords for each of these language pairs is likely to exceed the cost and time required to deploy a rule based transliteration model. The next chapter demonstrates the utility of a low precision rule based transliteration model for bootstrapping a statistical model that classifies words according to their etymological source.

⁴This is the section of the corpus with the highest percentage of Roman letter words (Riha and Baker, 2008a, b).

Chinese Gigaword Corpus (CNA 2004)

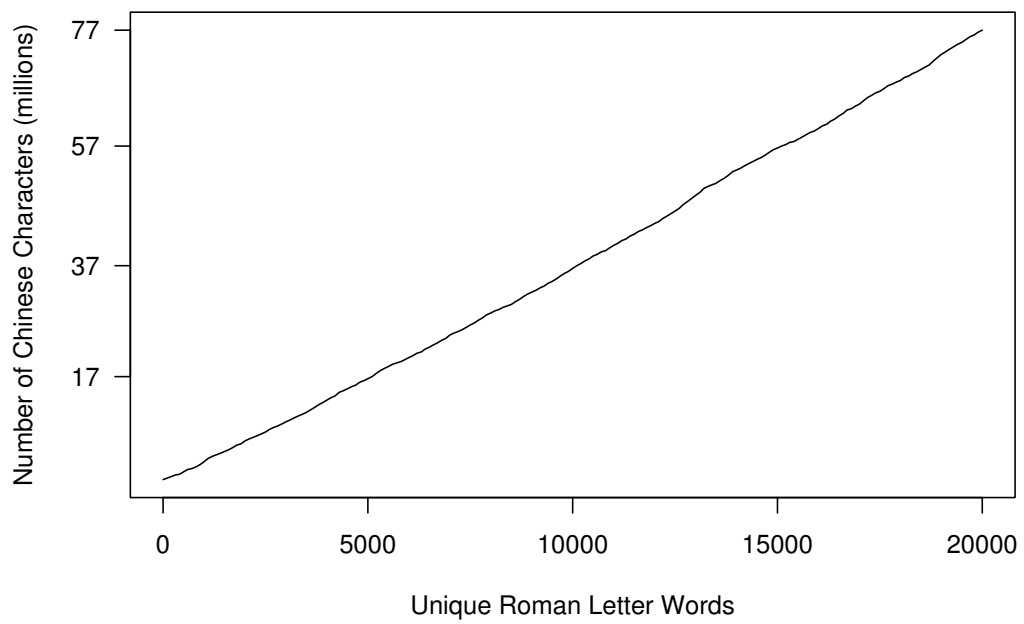


Figure 3.6: Number of unique Roman letter words by number of Chinese characters in the Chinese Gigaword Corpus (CNA 2004)