A weighted finite state transducer implementation of phoneme rewrite rules for English-to-Korean pronunciation conversion

Hahn Koo

San Jose State University, hahn.koo@sjsu.edu

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 202 – 208

Procedia Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

A Weighted Finite-State Transducer Implementation of Phoneme Rewrite Rules for English to Korean Pronunciation Conversion

Hahn Koo[a]*

[a]*San José State University, One Washington Square, San José, CA, 95192-0093, U.S.A*

Abstract

Words change their phonetic as well as orthographic form when they are borrowed and used by speakers of another language. A formal model that properly captures this change has theoretical implications in phonology and practical applications in speech processing and machine transliteration. This paper describes a method for developing a finite-state model that predicts how English words and named entities are pronounced in Korean. The model predicts nativized pronunciation using weighted finite-state transducers implementing context-dependent phoneme rewrite rules derived from English-to-Korean pronunciation pairs and syllable phonotactics in Korean.

Keywords: Pronunciation conversion; loanword phonology; transliteration; finte-state transducers

1. Introduction

A foreign word borrowed and used by speakers of another language sounds different from its original pronunciation. For example, /braɪ ər/ ('briar') in English is pronounced /pi ɾ ɑ iʌ / in Korean. It is worth developing a predictive model of the phenomenon for both theoretical and practical reasons. The phenomenon is called loanword phonology in linguistics and has been a topic of considerable research interest [1]. Foreign words form a major class of out-of-vocabulary words and pose problems for text-to-speech synthesis and automatic speech recognition. Many transliteration systems utilize cross-linguistic

* Hahn Koo. Tel.: +1-408-924-7093; fax: +1-408-924-4703.
E-mail address: hahn.koo@sjsu.edu.

doi:10.1016/j.sbspro.2011.10.599

phonetic correspondence to identify transliteration pairs or to predict the transliterated form of a given word [2], [3].

In this paper, I describe a method for developing a finite-state model that predicts how English words and named entities are pronounced in Korean. The model is essentially a composition of a set of weighted finite state transducers (WFST) that implement English-to-Korean phoneme rewrite rules and Korean syllable phonotactics. The rules are automatically derived from English-to-Korean transliteration pairs, possibly complementing rules manually written by language experts. The model can be implemented easily with publicly available toolkits for finite-state methods [4], [5], [6]. The rest of the paper is organized as follows. The basic structure of the proposed model is outlined in Section 2. Methods for deriving phoneme rewrite rules are described in Section 3. Results of experiments evaluating model performance are reported in Section 4, followed by a summary in Section 5.

2. Structure of the Proposed Model

The proposed model derives the Korean phoneme string that best matches a given English phoneme string by applying weighted rewrite rules to the constituent phonemes of the English string and choosing the best matching string that satisfies phonotactic constraints in Korean. This is done by solving

$$K\square = bestpath\ (E \circ R \circ P) \tag{1}$$

so as to find the FST representing the best matching Korean phoneme string $K\square$ by computing the best path within a WFST resulting from composing an FST *E* representing the given English phoneme string with a rewrite WFST *R* and a Korean phonotactic FST *P*.

2.1. Rewrite WFST

Following [7], a rewrite rule can be formulated as $\phi \rightarrow \psi \,/\, \lambda \,\_\_\, \rho$ to mean rewrite ϕ as ψ when preceded by λ and followed by ρ. In the proposed model, ϕ is a single English phoneme, ψ is a Korean phoneme string of length from zero to two, and λ and ρ can be a null symbol, a single English phoneme, or a word boundary marker. Put differently, a rewrite rule here defines how an English phoneme should be edited via deletion, substitution, or substitution followed by insertion depending on which phoneme appears to the left and/or right.

The model keeps one or more rewrite rules for every phoneme in English. Each expert rule specializing in a given English phoneme is weighted according to the probability with which the rule applies. See Section 3 below for details on how the rules are derived and their weights are estimated from data. During translation, the model maps each English phoneme in the input string to a Korean phoneme string – possibly more if the experts disagree – by applying the rewrite rules to one phoneme at a time.

The rewrite rules can be implemented as WFSTs [8]. Here, they are compiled into WFSTs using the GRM library [5]. Figure 1 illustrates a WFST implementing an example rule (2) with a weight of -0.36.

$$/p/_e \rightarrow /p^h/_k\ (-0.36)\ /\ \langle s \rangle\ \_\_\ /a/_e \tag{2}$$

In (2), phonemes subscripted with *e* and *k* respectively mean they are English phonemes and Korean phonemes. <*s*> is a word boundary marker labeling the beginning of a word. The number in the parentheses denotes the rule weight. In Figure 1, Σ denotes the input alphabet, in this case the set of

English phonemes plus boundary symbols <*s*> and </*s*>, which respectively mark word beginning and word end.

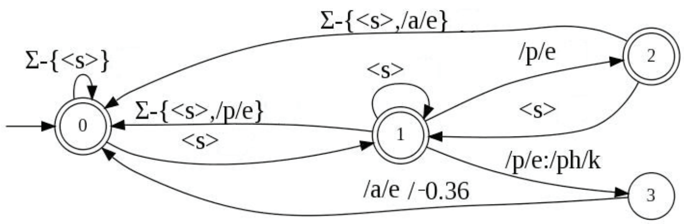

Fig. 1.A WFST representation of rule (2)

The set of expert rewrite rules for each English phoneme is basically a union of the corresponding WFSTs. While translating the whole input English phoneme string, the model applies a cascade of expert unions, one expert union for each English phoneme found in the input string. One can also apply a composition of expert unions to the input string in one fell swoop provided that the composed FST is not prohibitively large.

There is, however, a problem with this approach that needs a fix: applying a rule/WFST to translate an English phoneme may eliminate the context in which other rules apply. Caseiro et al. also faced the same problem while developing FSTs for grapheme-to-phoneme conversion [9]. As a solution, which I follow in this paper, they first redefine an input string by adding a place holder after every input symbol and then applying rewrite rules of the form

$$\phi\_ \rightarrow \phi\psi / \lambda' \_\_ \rho' \quad (3)$$

where _ after ϕ denotes a place holder which is replaced by ψ once the rule applies. The contexts λ' and ρ' in (3) refer to strings of input symbols skipping over place holders or what used to be place holders. The input symbols are removed after all the rules have applied to derive the final output string.

In sum, the function of the rewrite WFST R is to generate possible translations of input English phoneme string by first inserting place holders after every input phoneme, doing a phoneme-by-phoneme translation with rewrite rules of the form (3), and removing all English phonemes in each translation output. In other words, the structure of the rewrite WFST is such that

$$R = Ins \circ [E_1 \circ E_2 \circ \ldots \circ E_n] \circ Rm \quad (4)$$

where E_i denotes a union of WFSTs equivalent to expert rewrite rules for the i^{th} phoneme in English. *Ins* is an FST that inserts a place holder after every English phoneme in the input string. *Rm* is an FST that removes all English phonemes in the output string.

2.2. Korean phonotactic FST

In translating an English phoneme string, the rewrite WFST may suggest as possible translations phoneme strings that are not possible words in Korean. This can be prevented by composing the output of the rewrite WFST with an identity FST that accepts only phonotactically legitimate phoneme strings in Korean. Such a Korean phonotactic FST *P* is implemented here as

$$P = [<s>:<s>] \cdot S^{+} \cdot [</s>:</s>] \qquad (5)$$

where *S* is an identity FST that accepts phoneme strings that forms a possible syllable in Korean and + is the Kleene-plus operator. [<*s*>:<*s*>] and [</*s*>:</*s*>] are identity FSTs that respectively map the word boundary symbols <*s*> and </*s*> to themselves. The assumption is that a well-formed Korean word can be generated by choosing well-formed Korean syllables over one or more independent trials and concatenating them.

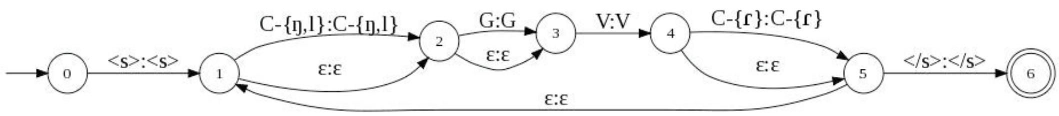

Fig. 2. An FST representation of Korean syllable phonotactics

The syllable FST *S* was manually compiled based on the fact that syllables in Korean follow a (C)(G)V(C) structure, where C, G, V, and parentheses respectively denote a non-glide consonant, a glide, a vowel, and optionality. Not all non-glide consonants in Korean can occupy both onset and coda: /l/ and /ŋ/ are only found in coda, whereas /ɾ / only occupies onset. This is illustrated in Figure 2, which shows the structure of the Korean phonotactic FST *P* built out of the syllable FST *S*.

3. Derivation of Rewrite Rules

3.1. Rule extraction via alignment

The rewrite rules that need to be derived are of the form $\phi \rightarrow \psi / \lambda \_\_ \rho$: the model needs to see tokens of ϕ:ψ pairs surrounded by λ and ρ in the database. It helps if the pronunciation pairs are segmented in terms of ϕ:ψ pairs or aligned in terms of units defining ϕ on one side and units defining ψ on the other. Recall that in this paper, ϕ is a single English phoneme and ψ is a Korean phoneme string of length from zero to two. An alignment WFST that recognizes all possible strings of ϕ:ψ pairs was created to this end. The WFST, called *align* below, has a topological structure schematically illustrated in Figure 3, where Φ:Ψ denotes the set of all possible ϕ:ψ pairs.

Each ϕ:ψ in *align* is a union of three WFSTs respectively for deletion [e:ϵ], substitution [e:k], and substitution plus insertion [e:k_1]·[ϵ :k_2], where e is an English phoneme, k, k_1, k_2 are single Korean phonemes, and ϵ is a null symbol. Each transition in those three WFTSs is initially weighted by the cost of substituting the input label with the output label. For substitution between two phonemes [e:k], the cost is one minus the feature similarity between the two phonemes computed following [10] in conjunction with the following binary features: consonantal, sonorant, continuant, voiced, nasal, strident, labial, coronal, dorsal, lateral, round, anterior, distributed, front, central, back, high, mid, low, advanced-tongue-root, spread glottis, and constricted glottis.

Substituting a phoneme with the null symbol [e:ϵ] and vice versa [ϵ :k] is assumed equally costly as substituting that phoneme with the second most similar phoneme in the other language. The idea is that deleting an English phoneme is not as desirable as replacing it by the most similar Korean phoneme but just as good as replacing it by second choice and likewise for inserting a Korean phoneme, which is the same as deleting that Korean phoneme from the opposite perspective.

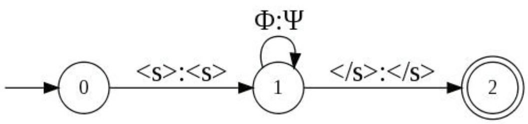

Fig.3. Topological structure of *align*

The weights can be trained on a given dataset using the EM algorithm [11]. For the model in this paper, the weights of *align* were trained using the MIT finite-state transducer toolkit [6]. The end result is a WFST over the tropical semiring which computes the best alignment π between a given pronunciation pair in terms of ϕ:ψ pairs by

$$\pi = bestpath\ (E \circ align \circ K) \qquad (6)$$

where E and K denote FSTs representing English and Korean pronunciations in the given pair. The alignment suggests the most likely sequence of context-*independent* rewrite rules of the form $\phi \rightarrow \psi$ applied to derive one pronunciation from the other. One can simply add the English phonemes to the left and right of ϕ to make the rules context dependent. Basically, after alignment is complete, the model is collecting trigrams of English phonemes and remembering how the phoneme in the center is rewritten.

As mentioned in 2.1, each rule thus extracted is compiled into an equivalent WFST. All rules that affect the same English phoneme are grouped together and the corresponding transducers are combined into an expert transducer via union operation.

3.2. Rule weighting and back-off

The individual rules are weighted by their negated maximum likelihood estimate conditional probabilities: minus one times how often a rule is observed to rewrite a given English phoneme in training data normalized by the sum of such frequencies over all rules targeting the same phoneme. One could think of the weight of a rule as the cost of not applying the rule: not applying a more likely rule is worse than not applying a less likely rule. The finite-state toolkits used here by default assume the tropical semiring defined over negated log probabilities rather than negated probabilities. However, using the negated probabilities instead pose little problem for the important operations – composition, union, and best-path – necessary to construct the current model.

Recall that the rule derivation process after alignment is none other than counting trigrams in paired data. One must properly deal with the data sparsity problem that usually follows models that use higher order n-grams. However, as the focus of this study is to develop a prototype WFST implementation, the model relies on the following heuristics: input phonemes are first processed by all context-dependent rules known to the model and those that are left uncovered are processed by context independent rules at the end.

4. Evaluation

The proposed model was trained and tested using a dataset consisting of 5,812 English-Korean pronunciation pairs chosen from a lexicon compiled by the National Institute of the Korean Language [12]. The lexicon lists foreign words in their original spelling form along with their transliterated form in the Korean writing system. English pronunciation of each word was transcribed by looking up the CMU pronunciation dictionary, while the Korean pronunciation was transcribed by applying a handwritten set of letter-to-sound rules to the corresponding transliterated form in Korean. Lexical stress information was ignored in the transcription process.

The 5,812 pairs were randomly split into a training set and a test set in a four-to-one ratio. The model was developed on 4,649 pronunciation pairs and predicted the Korean pronunciation of English words in the remaining 1,163 pairs. Performance of the model was evaluated in terms of word accuracy: a prediction was deemed correct only if the 1-best output of the model was identical to the form found in the data.

As shown in Table 1, the model correctly predicts 55.80% of the pairs found in the test data. To see how much contextual information boosts performance, a model that rewrites phonemes using the context-independent rules in 3.1 alone was also evaluated. About 15.65% of the pairs cannot be handled properly without contextual information.

Table 1. Performance of the proposed model

Model	Word accuracy
Context-independent rules	40.15%
+ Context-dependent rules	55.80%

The performance of the model seems comparable to the results from previous English-to-Korean transliteration studies. Jung et al. [13], for example, proposes a statistical transliteration model whose 1-best output shows a mean word accuracy of 54.9% in a 10-fold cross validation experiment using a list of 8,368 English-Korean word pairs.

5. Conclusion

I presented a model that predicts how pronunciations of English words and named entities are nativized in Korean using WFSTs representing context-dependent phoneme rewrite rules derived from English-to-Korean pronunciation pairs and syllable phonotactics in Korean. The model can be easily created using well-known methods and toolkits in the finite-state literature and performs at a level comparable to that of other statistical models for English-to-Korean transliteration in previous studies.

References

[1] Silverman D. Multiple scansions in loanword phonology: evidence from Cantonese. *Phonology* 1992; **9**: 289-328.

[2] Knight K, Graehl J. Machine transliteration. *Computational Linguistics* 1998; **24**: 599-612.

[3] Tao T, Yoon S, Fister A, Sproat R, Zhai C. Unsupervised named entity transliteration using temporal and phonetic correlation. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*; 2006, p. 250-257.

[4] Mohri M, Pereira FCN, Riley M. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science* 2000; **231**: 17-32.

[5] Mohri M. Weighted grammar tools: the GRM library. In: *Robustness in Language and Speech Technology,* The Netherlands: Kluwer Academic Publishers; 2001, p. 165-186.

[6] Hetherington L. The MIT finite-state transducer toolkit for speech and language processing. In: *Proceedings of the 8th International Conference on Spoken Language Processing*; 2004, p. 2609-2612.

[7] Kaplan R, Kay M. Regular models of phonological rule systems. *Computational Linguistics* 1994; **20**: 331-378.

[8] Mohri M, Sproat R. An efficient compiler for weighted rewrite rules. In: *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*; 1996, p. 231-238.

[9] Caseiro D, Trancoso I, Oliveira L, Viana C. Grapheme-to-phoneme using finite-state transducers. In: *Proceedings of the 2002 IEEE Workshop on Speech Synthesis*; 2002.

[10] Frisch SA, Pierrehumbert JB, Broe M. Similarity avoidance and the OCP. *Natural Language and Linguistic Theory* 2004; **22**: 179-228.

[11] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 1977; **39**: 1-21.

[12] NIKL. Survey of the state of loanword usage. An Electronic Document by the National Institute of the Korean Language; 2008. Available at http://www.korean.go.kr/09_new/include/Download.jsp?path=OpenPds&sub=1&idx=105.

[13] Jung SY, Hong S, Paek E. An English to Korean transliteration model of extended Markov window. In: *Proceedings of the 18th International Conference on Computational Linguistics*; 2000, p. 383-389.