Grapheme-to-Phoneme Conversion Using Automatically Extracted Associative Rules for Korean TTS System

Jinsik Lee, Seungwon Kim and Gary Geunbae Lee

Department of Computer Science and Engineering
Pohang University of Science and Technology, South Korea
{palcery, rockzja, gblee}@postech.ac.kr

ABSTRACT

In this paper, we describe a method for automatically extracting grapheme-to-phoneme conversion rules directly from the transcription of speech synthesis database and introduce a weighted score and *jamo* * similarity to overcome the rule application difficulties. We make a structured rule tree by rule pruning and rule association, and can eliminate most of the rules with almost no decrease of the performance. Our system achieves over 99.5 percent of phoneme-level accuracy and this performance is easily achievable even with the small amount of training data.

Index Terms: grapheme-to-phoneme conversion, letter-to-sound rule, text-to-speech system

1. INTRODUCTION

Grapheme-to-phoneme (G2P) conversion is one of the essential components of text analyzer in text-to-speech (TTS) system. It is also used to create a pronunciation dictionary for speech recognizer. The problem of G2P conversion is figuring out how to transform a given text to some predefined phonetic symbols. Similar to other languages, the G2P mapping of Korean language is a complex process since not all the graphemes are realized and some are mapped to different phonemes depending on the context.

Several methods and approaches have been proposed for G2P conversion problems. One approach is using a hybrid system composed of regular rules and exception dictionaries [1, 2, 3]. Since regular pronunciation rules for Korean language can be easily implemented compared to other languages, usually, they have been written by hand, which has relatively good conversion accuracy. These facts have led researches to focus on the study of exceptional cases [4]. Another approach is extracting conversion rules from corpora [5, 6, 7]. These automatically produced conversion rules are usually in the form of context-sensitive two-level rules.

In concatenation-based TTS system, unit selection is one of the most important parts for making natural and high-quality speech. Thus, the phonetic symbols used in unit selection procedure must conform to the transcription of the synthesis speech database (DB). In other words, G2P for TTS system should extract conversion rules from the transcription of the synthesis speech DB so that the phonetic symbols of the units used in unit selection procedure are fully compatible to the corresponding phonemes.

Our G2P system also extracts two-level rules similar to previous works [5, 6, 7], but merges the rules which are associable. Then, the readability of each rule is increased, and the time spent during the converting process is reduced, since the rule chunks become larger.

Our system is composed of three rule dictionaries (onset, nucleus, and coda) and simple modules for G2P conversion, instead of extra exception dictionaries and complex modules to reflect the regular rules and various features (POS tag, word position, articulatory-phonetic features, phonological features, etc.). By extracting the rules even from the exceptional cases, it is not necessary to store all of the cases any more. Instead, it is enough to simply keep their abstract patterns in our system.

This paper is organized as follows. In section 2, we will describe the procedures of rule generation including alignment, rule pruning, and rule association. In section 3, we will describe the main module (G2P conversion) with a method to deal with rule conflict and use of *jamo* similarity. In section 4, we will describe the corpora that have been used along with experimental results. Finally, in section 5, we will give a summary.

2. G2P RULE GENERATION

Our system automatically extracts the G2P conversion rules directly from the transcription of synthesis speech DB. They are in the form of rewrite rules as follows:

$$r\text{: } L(G)R \rightarrow P$$

The rule r means that a given set of graphemes G can be transformed to a set of phonemes P, when G occurs in the left contexts of string L and the right contexts of string R. The contexts L or R include information of adjacent graphemes and *eojeol*† boundaries and sentence boundaries. Both G and P can be composed only of graphemic or phonemic null symbols due to insertion or deletion.

* *Jamo*: a set of consonants and vowels used in Korean. The word *jamo* is derived from *ja*, which means consonant, and *mo*, which means vowel. Generally two or three *jamos* make up a syllable.

† *Eojeol*: a sequence of one or more syllables, separated by spaces. An *eojeol* usually consists of one or more stem morphemes and functional morphemes.

2.1 Alignment

Korean syllables consist of three components: an initial consonant (onset), a vowel (nucleus), and a final consonant (coda). However, the final consonant in graphemes or phonemes and the initial consonant in phonemes can be omitted. To make each syllable into the canonical form (onset-nucleus-coda triple), we added the graphemic or phonemic null symbols ('_'). Then, the alignment process becomes straightforward as in Figure 1.

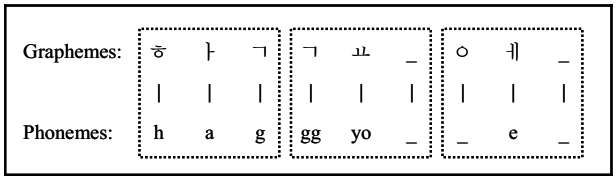

Figure 1: Alignment of graphemes and phonemes in Korean

2.2 Rule extraction

After the alignment procedure, we can easily perform the automatic rule extraction. In this step, we extract all possible rules whose context length is from 2 to 6 in order to give the limitation on the number of extracted rules in rule dictionaries.

We have found that the phonetic changes are quite different along the position of *jamo* in a syllable. For this reason, we kept each extracted rule in three different rule dictionaries according to the *jamo* positions (onset, nucleus, and coda).

Under the condition of the limitation in context length, the phonemes corresponding to the given graphemes may be differently realized although the given graphemes are identical. In this case, we first extracted such rules, and then counted the number of each candidate phoneme, and finally computed the realization probability of the specific phoneme *p* as follows:

$$\Pr(p \mid L(G)R) = \frac{Count(L(G)R \rightarrow p \in P)}{Count(L(G)R \rightarrow P)}$$

This realization probability will be used for resolving the rule conflicts (see section 3.1).

2.3 Pruning

In the rule extraction procedure, since all of the possible rules in a certain context length are extracted, the size of the rule dictionaries becomes quite huge. It is essential to prune the rules in order to form smaller rule dictionaries and to structure the unstructured rule entries.

The candidate rule to be pruned should satisfy the following two conditions: (1) there exists a parent rule which has shorter contexts and (2) the set of phoneme *P* of the parent rule contains only one candidate phoneme. The pruning process is shown in Figure 2.

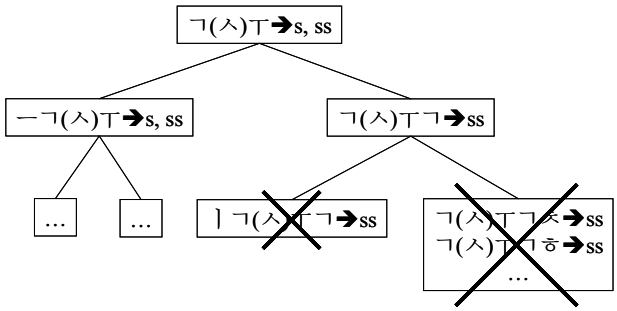

Figure 2: An example of the possible rule pruning

Then, we can form the structured rule trees using parent-children relationship according to the different contexts. One possible rule tree is given in Figure 3. In Figure 3, '*' and '+' in left side of the rules represent a sentence boundary and an *eojeol* boundary, respectively.

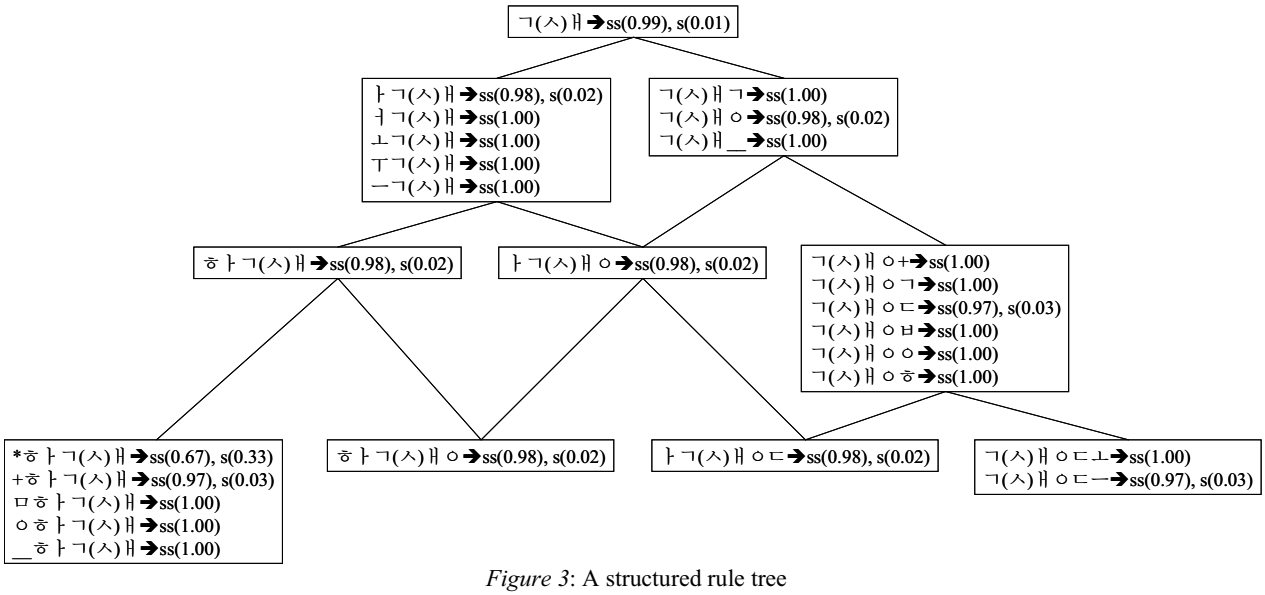

Figure 3: A structured rule tree

2.4 Rule association

Until now, we have only considered the special case in which the length of each grapheme and phoneme is one. However, we can merge the rules to make associative rules as shown in Figure 4.

The rule association can be performed only under the following conditions: (1) the contexts of each candidate rule should be the same so that the context of the associative rule has the same coverage and (2) the set of phoneme *P* of each candidate rule contains only one phoneme. In other words, we do not allow the associative rules to be generated from multiple different phonemes.

The rule association increases the readability of the conversion rules, and decreases the size of the rule dictionaries so that we can save time to search the entries in the rule dictionaries.

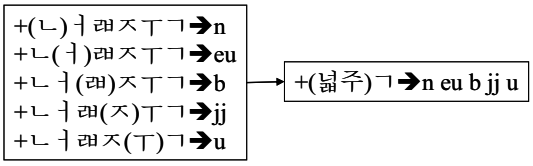

Figure 4: An example of the possible rule association

3. G2P CONVERSION

To obtain the phonemes corresponding to the given graphemes, we first transform the graphemes to the canonical form as in the alignment procedure. Then, we can directly apply the automatically extracted conversion rules. However, there are two issues to handle on this direct conversion process.

3.1 Resolving the rule conflicts

When we apply the conversion rules, there exist several applicable rules, since every parent rule of the applicable rule is also applicable. The one way to resolve the rule conflicts is simply applying the most specific rule and choosing the phoneme with the highest realization probability. However, this method gives the phoneme which is over-fitted to the training corpus. Generally, the most specific rules represent the exceptional cases, so we need to adjust the degree of figuring out exceptional cases with the score.

The score is computed by the summation of the weighted realization probability as follows:

$$Score(p \mid L(G)R) = \sum w_{L',R'} \Pr(p \mid L'(G)R')$$

where $w_{L',R'}$ is a weight determined by the length of the contexts, and *L'* and *R'* are the contexts which are shorter than or equal to *L* and *R*, i.e., the rule *L'(G)R'*→*P* is the parent rule of the rule *L(G)R*→*P* or the rule *L(G)R*→*P* itself. In our system, we set the most specific rules to have the highest weight. An example of calculating the score of each candidate phoneme is given in Figure 5.

Given grapheme: ㅅㅣㄴ(ㄱ)ㅗ_ㅎ

<Applicable rules in the dictionary>
Context length 2: ㄴ(ㄱ)ㅗ➔g(0.88), gg(0.12)
Context length 3: ㄴ(ㄱ)ㅗ_➔g(0.78), gg(0.22), ㅣㄴ(ㄱ)ㅗ➔g(0.86), gg(0.14)
Context length 4: ㄴ(ㄱ)ㅗ_ㅎ➔g(1.00), ㅣㄴ(ㄱ)ㅗ_➔g(0.50), gg(0.50),
ㅅㅣㄴ(ㄱ)ㅗ➔g(0.60), gg(0.40)

Score (g | ㅅㅣㄴ(ㄱ)ㅗ_ㅎ) = $w_{1,1}$ pr (g | ㄴ(ㄱ)ㅗ) + $w_{1,2}$ pr (g | ㄴ(ㄱ)ㅗ_)
+ $w_{2,1}$ pr (g | ㅣㄴ(ㄱ)ㅗ) + $w_{1,3}$ pr (g | ㄴ(ㄱ)ㅗ_ㅎ)
+ $w_{2,2}$ pr (g | ㅣㄴ(ㄱ)ㅗ_) + $w_{3,1}$ pr (g | ㅅㅣㄴ(ㄱ)ㅗ)
= 2(0.88) + 3(0.78) + 3(0.86) + 4(1.00) + 4(0.50) + 4(0.60) = 15.08

Score (gg | ㅅㅣㄴ(ㄱ)ㅗ_ㅎ) = $w_{1,1}$ pr (gg | ㄴ(ㄱ)ㅗ) + $w_{1,2}$ pr (gg | ㄴ(ㄱ)ㅗ_)
+ $w_{2,1}$ pr (gg | ㅣㄴ(ㄱ)ㅗ) + $w_{1,3}$ pr (gg | ㄴ(ㄱ)ㅗ_ㅎ)
+ $w_{2,2}$ pr (gg | ㅣㄴ(ㄱ)ㅗ_) + $w_{3,1}$ pr (gg | ㅅㅣㄴ(ㄱ)ㅗ)
= 2(0.12) + 3(0.22) + 3(0.14) + 4(0.00) + 4(0.50) + 4(0.40) = 4.92

Figure 5: An example of calculating the score of the phonemes

3.2 Coping with data sparseness

It is possible that no applicable rule exists due to data sparseness problem. In this case, we can apply the rule with the target phoneme which has the most similar contexts to the given contexts including the graphemes. To do so, we have defined *jamo* similarity as follows:

$$Similarity(j_1, j_2) = \sum_{f \in F} \alpha_f g_f(j_1, j_2)$$

The function g_f gives 1 if two *jamos* j_1 and j_2 have the same property (+/-) of distinctive articulatory-phonetic and phonological feature *f*, and otherwise gives 0. The coefficient α_f plays a role of weight for each feature. The set of distinctive features *F* consists of the following 14 features: vocalic, consonantal, oral, anterior, coronal, continuant, strident, nasal, glottalized, aspirated, back, high, low, and round [8].

We set the minimum context length of each rule to be greater than two to guarantee more effective rule application, which is different from the previous works [7]. It means that the rule must have at least one left context and one right context. If there is a rule which does not have any context, i.e., *L* and *R* is an empty set, then the rule always gives the phoneme with the highest realization probability without pertaining to the contexts.

4. EXPERIMENTAL RESULTS

4.1 Corpora used

The corpora used in this research are the transcriptions of two speech synthesis DB. One is a transcription of our own reading-style speech DB, and the other is a transcription of conversational-style speech DB from ETRI (Electronics and Telecommunications Research Institute). Our own reading-style speech DB is a balanced corpus, and it contains various patterns of phonetic changes. The statistic of the corpora used is given in Table 1.

Table 1: The statistic of the corpora used

	# of sentences	# of *eojeols*	# of diff. *eojeols*
Reading-style	4,700	64,263	38,077
Conversational-style	2,008	8,941	3,619

4.2 Performance evaluation

The two corpora consist of the pairs of sentences (graphemes) and transcriptions (phonemes). We first aligned them and extracted all the possible rules which satisfy the condition of the context length. Then, we made the set of the rules to the structured form and applied the rule association (RA) and the rule pruning (PR). The experiments of the evaluating transcription accuracy are given in Table 2.

Table 2: Transcription accuracy on reading-style DB and conversational-style DB

	Reading-style		Conversational-style	
	Accuracy	# rules	Accuracy	# rules
Full rules	99.595%	2,005K	99.559%	252K
+RA	99.562%	1,003K	99.541%	117K
+RA+PR	99.560%	35K	99.508%	5K

The results on Table 2 are phoneme-level accuracies obtained by 5-fold cross-validation. The "full rules" means applying all the rules without any rule association and pruning. After the rule pruning and the rule association, it shows that the most of the rules are eliminated with almost no decrease of the accuracy. This performance is close to the state-of-the-art level when we compare with the performances of other similar approaches [6, 7] even though direct performance comparison is not possible due to the different experiment environments.

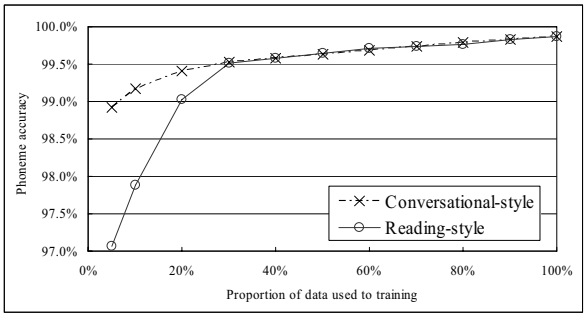

Figure 6: Accuracy along the proportion of the training data

Figure 6 shows the performance saturation curves of our system. In this experiment, we did not separate the training data and the test data. Instead, we used the whole data as the test data and some proportion of the data as the training data. Theoretically, if there is no limitation of the context length and the whole data is used as the training data, then the 100 percent accuracy will be achieved. Although the actual situation is under the limitation of the context length, it achieved 99.9 percent accuracy. Surprisingly, it requires only 30 percent of the whole data to reach the 99.5 percent of the accuracy, which explains why we didn't have a performance decrease in table 2 even with massive decrease of the number of the rules by using the rule association and rule pruning.

5. CONCLUSION

We have described a method for automatically extracting G2P conversion rules from a domain-dependent transcription of speech synthesis DB. Since the phonemes generated by the G2P conversion module should be compatible to the phonetic symbols used in unit selection procedure, it is necessary to conform the phonemes to the transcription in a concatenation-based TTS system.

We have introduced weighted scores and *jamo* similarity, and obtained over 99.5 percent of phoneme-level accuracy. Most of the rules can be eliminated by rule pruning and rule association with almost no decrease of the performances.

For future works, we plan to apply this method to other problems such as spelling correction and study the usefulness of domain-adaptable G2P in unit selection.

6. ACKNOWLEDGEMENTS

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2005-(C1090-0501-0018)).

7. REFERENCES

[1] K.-N. Lee, M. Chung, "Automatic Generation of Pronunciation Variants for Korean Continuous Speech Recognition," *The Journal of the Acoustical Society of Korea*, Vol. 20, No. 2: 35-43, 2001.

[2] B. Kim, G. G. Lee, J.-H. Lee, "Morpheme-Based Grapheme to Phoneme Conversion Using Phonetic Patterns and Morphophonemic Connectivity Information," *ACM Transactions on Asian Language Information Processing*, Vol. 1, No. 1: 65-82, 2002.

[3] S. Kim, J.-E. Ahn, S.-H. Kim, Y.-H. Lee, "A Korean Grapheme-to-Phoneme Conversion System Using Selection Procedure for Exceptions", In *Proceedings of ICSLP-04*, 1905-1908, 2004.

[4] S. Kim, "Phonology of Exceptions for Korean Grapheme-to-Phoneme Conversion," In *Proceedings of ICSLP-04*, 1285-1289, 2004.

[5] J. H. Jeon, M. Chung, "Automatic Generation of Domain-Dependent Pronunciation Lexicon with Data-Driven Rules and Rule Adaptation," In *Proceedings of EUROSPEECH-05*, 1337-1340, 2005.

[6] P. Massimino, A. Pacchiotti, "An Automaton-Based Machine Learning Technique for Automatic Phonetic Transcription," In *Proceedings of EUROSPEECH-05*, 1901-1904, 2005.

[7] A. Chalamandaris, S. Raptis, P. Tsiakoulis, "Rule-Based Grapheme-to-Phoneme Method for the Greek," In *Proceedings of EUROSPEECH-05*, 2937-2940, 2005.

[8] M.-O. Choi, *Korean Phonology*, Taehaksa, 2004 (written in Korean).