Decision-Tree based Error Correction for Statistical Phrase Break Prediction in Korean *

Byeongchang Kim and **Geunbae Lee**
Department of Computer Science & Engineering
Pohang University of Science & Technology
Pohang, 790-784, South Korea
{bckim, gblee}@postech.ac.kr

Abstract

In this paper, we present a new phrase break prediction architecture that integrates probabilistic approach with decision-tree based error correction. The probabilistic method alone usually suffers from performance degradation due to inherent data sparseness problems and it only covers a limited range of contextual information. Moreover, the module can not utilize the selective morpheme tag and relative distance to the other phrase breaks. The decision-tree based error correction was tightly integrated to overcome these limitations.
The initially phrase break tagged morpheme sequence is corrected with the error correcting decision tree which was induced by C4.5 from the correctly tagged corpus with the output of the probabilistic predictor. The decision tree-based post error correction provided improved results even with the phrase break predictor that has poor initial performance. Moreover, the system can be flexibly tuned to new corpus without massive retraining.

1 Introduction

During the past few years, there has been a great deal of interest in high quality text-to-speech (TTS) systems (van Santen et al., 1997). One of the essential problems in developing high quality TTS systems is to predict phrase breaks from texts. Phrase breaks are especially essential for subsequent processing in the TTS systems such as grapheme-to-phoneme conversion and prosodic feature generation. Moreover, graphemes in the phrase-break boundaries are not phonologically changed and should be pronounced as their original corresponding phonemes.

There have been two approaches to predict phrase breaks (Taylor and Black, 1998). The first uses some sort of syntactic information to predict prosodic boundaries based on the fact that syntactic structure and prosodic structure are co-related. This method needs a reliable parser and syntax-to-prosody module. These modules are usually implemented in rule-driven methods, consequently, they are difficult to write, modify, maintain and adapt to new domains and languages. In addition, a greater use of syntactic information will require more computation for finding a more detailed syntactic parse. Considering these shortcomings, the second approach uses some probabilistic methods on the crude POS sequence of the text, and this method will be further developed in this paper. However, the probabilistic method alone usually suffers from performance degradation due to inherent data sparseness problems.

So we adopted decision tree-based error correction to overcome these training data limitations. Decision tree induction is the most widely used learning method. Especially in natural language and speech processing, decision tree learning has been applied to many problems including stress acquisition from texts, grapheme to phoneme conversion and prosodic phrase modeling (Daelemans et al., 1994) (van Santen et al., 1997) (Lee and Oh, 1999).

In the next section, linguistic features of Korean relevant to phrase break prediction are described. Section 3 presents the probabilistic phrase break prediction method and the tree-based error correction method. Section 4 shows experimental results to demonstrate the performance of the method and section 5 draws some conclusions.

2 Features of Korean

This section briefly explains the linguistic characterists of spoken Korean before describing the phrase break prediction.

1) A Korean word consists of more than one morpheme with clear-cut morpheme boundaries (Korean is an agglutinative language).

* This paper was supported by the University Research Program of the Ministry of Information & Communication in South Korea through the IITA(1998.7-2000.6).

2) Korean is a postpositional language with many kinds of noun-endings, verb-endings, and prefinal verb-endings. These functional morphemes determine a noun's case roles, a verb's tenses, modals, and modification relations between words. 3) Korean is basically an SOV language but has relatively free word order compared to other rigid word-order languages such as English ,except for the constraints that the verb must appear in a sentence-final position. However, in Korean, some word-order constraints actually do exist such that the auxiliary verbs representing modalities must follow the main verb, and modifiers must be placed before the word (called head) they modify. 4) Phonological changes can occur in a morpheme, between morphemes in a word, and even between words in a phrase, but not between phrases.

3 Hybrid Phrase Break Detection

Part-of-speech (POS) tagging is a basic step to phrase break prediction. POS tagging systems have to handle out-of-vocabulary (OOV) words for an unlimited vocabulary TTS system. Figure 1 shows the architecture of our phrase break predictor integrated with the POS tagging system. The POS tagging system employs generalized OOV word handling mechanisms in the morphological analysis and cascades statistical and rule-based approaches in the two-phase training architecture for POS disambiguation.

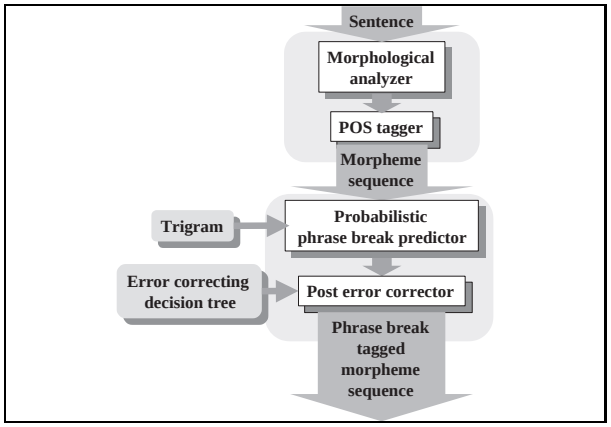

Figure 1: Architecture of the hybrid phrase break prediction.

The probabilistic phrase break predictor segments the POS sequences into several phrases according to word trigram probabilities. The initial phrase break tagged morpheme sequence is corrected with the error correcting tree learned by the C4.5 (Quinlan, 1983).

The next two subsections will give detailed descriptions of the probabilistic phrase prediction and error correcting tree learning. The hybrid POS tagging system will not be explained in this paper, and the interested readers can see (Cha et al., 1998) for further reference.

3.1 Probabilistic Phrase Break Detection

3.1.1 Probabilistic Models

For phrase break prediction, we develop the word POS tag trigram model. Some experiments are performed on all the possible trigram sequences and 'word-tag word-tag *break* word-tag' sequence turns out to be the most fruitful of any others, which are the same results as the previous studies in English (Sanders, 1995).

The probability of a phrase break b_i appearing after the second word POS tag is given by

$$P(b_i|t_1t_2t_3) = \frac{C(t_1t_2b_it_3)}{\sum_{j=0,1,2} C(t_1t_2b_jt_3)},$$

where C is a frequency count function and b_0, b_1 and b_2 mean no break, minor break and major break, respectively. Even with a large number of training patterns it is very clear that there will be a number of word POS tag sequences that never occur or occur only once in the training corpus. One solution to this data sparseness problem is to smooth the probabilities by using the bigram and unigram probabilities, which adjusts the frequency counts of rare or non-occurring POS tag sequences. We use the smoothed probabilities:

$$\begin{aligned} P(b_i|t_1t_2t_3) &= \lambda_1 \frac{C(t_1t_2b_it_3)}{\sum_{j=0,1,2} C(t_1t_2b_jt_3)} \\ &+ \lambda_2 \frac{C(t_2b_it_3)}{\sum_{j=0,1,2} C(t_2b_jt_3)} \\ &+ \lambda_3 \frac{C(t_2b_i)}{\sum_{j=0,1,2} C(t_2b_j)}, \end{aligned}$$

where λ_1, λ_2 and λ_3 are three nonnegative constants such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. In some experiments, we can get the weights λ_1, λ_2 and λ_3 as 0.2, 0.7 and 0.1, respectively.

3.1.2 Adjusting the POS Tag Sequences of Words

Previous researchers of phrase break prediction used mainly content-function word rule, whereby a phrase break is placed before every function word that follows a content word (Allen and Hunnicut, 1987) (Taylor et al., 1991). The

researchers used tag set size of only 3, including function, content and punctuation in the rule.

However, Korean is a post-positional agglutinative language. If the content-function word rule is to be adapted in Korean, the rule must be changed so that a phrase break is placed before every content morpheme that follows a function morpheme. Unfortunately this rule is very inefficient in Korean since it tends to create too many pauses. In our works, only the POS tags of function morphemes are used because the function morphemes constrain the classes of precedent morphemes and play important roles in syntactic relation. So, each word is represented by the POS tag of its function morpheme. In the case of the word which has no function morpheme, simplified POS tags of content morphemes are used. The number of POS tags used in this research is 32.

3.2 Decision-Tree Based Error Correction

The probabilistic phrase break prediction only covers a limited range of contextual information, i.e. two preceding words and one following word. Moreover, the module can not utilize the morpheme tag selectively and relative distance to the other phrase breaks. For this reason we designed error correcting tree to compensate for the limitations of the probabilistic phrase break prediction. However, designing error correcting rules with knowledge engineering is tedious and error-prone. Instead, we adopted decision tree learning approach to automatically learn the error correcting rules from a correctly phrase break tagged corpus.

Most algorithms that have been developed for building decision trees employ a top-down, greedy search through the space of possible decision trees (Mitchell, 1997). The C4.5 (Quinlan, 1983) is adequate to build a decision tree easily for successively dividing the regions of feature vector to minimize the prediction error. It also uses information gain which measures how well a given attribute separates the training vectors according to their target classification in order to select the most critical attributes at each step while growing the tree (hence the name is IG-Tree). Now, we utilize it for correcting the initially phrase break tagged POS tag sequences generated by probabilistic predictor.

However, we invented novel way of using the decision tree as transformation-based rule induction (Brill, 1992). Figure 2 shows the tree learning architecture for phrase break error correction. The initial phrase break tagged POS tag sequences support the feature vectors for attributes which are used for decision making. Because the feature vectors include phrase break sequences as well as POS tag sequences, a learned decision tree can check the morpheme tag selectively and utilize the relative distance to the other phrase breaks. The correctly phrase break tagged POS tag sequences support the classes into which the feature vectors are classified. C4.5 builds a decision tree from the pairs which consist of the feature vectors and their classes.

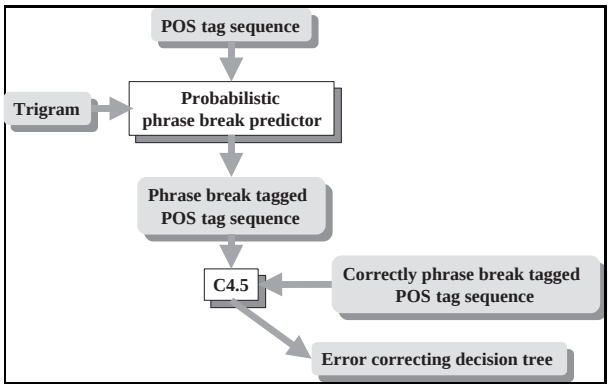

Figure 2: Architecture of the error correcting decision tree learner.

4 Experimental Results

4.1 Corpus

The experiments are performed on a Korean news story database, called MBCNEWSDB, of spoken Korean directly recorded from broadcasting news. The size of the database is now 6,111 sentences (75,647 words) and it is continuously growing. To be used in the phrase break prediction experiments, the database has been POS tagged and break-labeled with major and minor phrase breaks.

4.2 Phrase Break Detection and Error Correction

We performed three experiments to show synergistic results of probabilistic method and tree-based error correction method. First, only probabilistic method was used to predict phrase breaks. Trigrams, bigrams and unigrams for phrase break prediction were trained from the break-labeled and POS tagged 5,492 sentences of the MBCNEWSDB by adjusting the POS sequences of words as described in subsection 3.1.2. The other 619 sentences are used to test the performance of the probabilistic phrase break predictor. In the second experiment, we made a decision tree, which can be used only to predict phrase breaks and cannot be used to

correct phrase breaks, from the 5,429 sentences. Also the 619 sentences were used to test the performance of the decision tree-based phrase break predictor. The size of feature vector (the size of the window) is varied from 7 (the POS tag of current word, preceding 3 words and following 3 words) to 15 (the POS tag of current word, preceding 7 words and following 7 words). The third experiment utilized a decision tree as post error corrector as presented in this paper. We trained trigrams, bigrams and unigrams using 60% of total sentences, and learned the decision tree using 30% of total sentences. For the other experiment, 50% and 40% of total sentences are used for probability training and for decision tree learning, respectively. The other 10% of total sentences were used to test as in the previous experiments(Figure 3). For the decision tree in the third experiment, though the size of the window is also varied from 7 words to 15 words, the size of feature vector is varied from 14 to 30 because phrase breaks tagged by probabilistic predictor are include in the feature vector.

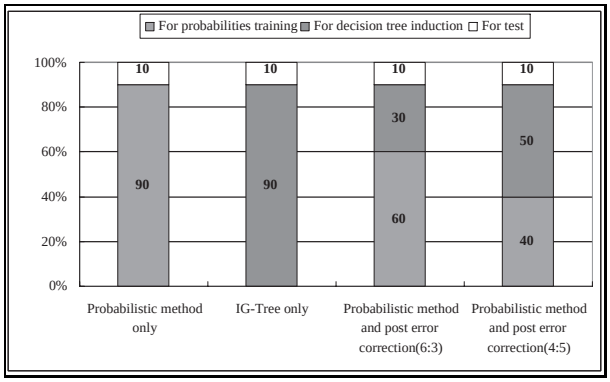

Figure 3: The number of sentences for the probability training, the decision tree learning and the test in the experiments.

The performance is assessed with reference to N, the total number of junctures (spaces in text including any type of phrase breaks), and B, the total number of phrase breaks (only minor(b_1) and major(b_2) breaks) in the test set. The errors can be divided into insertions, deletions and substitutions. An insertion (I) is a break inserted in the test sentence, where there is not a break in the reference sentence. A deletion (D) occurs when a break is marked in the reference sentence but not in the test sentence. A substitution (S) is an error between major break and minor break or vice versa. Since there is no single way to measure the performance of phrase break prediction, we use the following performance measures (Taylor and Black, 1998).

$$Break\_Correct = \frac{B - D - S}{B} \times 100\%,$$

$$Juncture\_Correct = \frac{N - D - S - I}{N} \times 100\%$$

We use another performance measure, called *adjusted score*, which refer to the prediction accuracy in proportion to the total number of phrase breaks as following performance measure proposed by Sanders (Sanders, 1995).

$$Adjusted\_Score = \frac{JC - NB}{1 - NB},$$

where NB[1] means the proportion of no breaks to the number of interword spaces and JC means the *Juncture_Correct/100*.

Table 1 shows the experimental results of our phrase break prediction and error correction method on the 619 open test sentences (10% of the total corpus). In the table, W means the feature vector size for the decision tree, and 6:3 and 4:5 mean ratio of the number of sentences used in the probabilistic train and the decision tree induction.

The performance of probabilistic method is better than that of IG-tree method with any window size in $Break\_Correct$. However, as the feature vector size is growing in IG-tree method, $Juncture\_Correct$ and $Adjusted\_Score$ become better than those of the probabilitic method. From the fact that the attribute located in the first level of the decision trees is the POS tag of preceding word, we can see that the POS tag of preceding word is the most useful attribute for predicting phrase breaks.

The performance before the error correction in hybrid experiments is worse than that of the original probabilistic method because the size of training corpus for probabilistic method is only 66.6% and 44.4% of that of the original one, respectively. However, the performance sets improved by the post error correction tree, and becoms finally higher than that of both the probabilistic method and the IG-tree method. The attribute located in the first level of the decision tree is the phrase break that was predicted in the probabilistic method phase. Although the initial performance (before error correction) of the experiment using 4:5 corpus ratio is worse than that of the experiment using 6:3 corpus ratio, the final performance gets impressively improved as the decision tree induction corpus

[1] $NB = \frac{N-B}{N}$

Table 1: Phrase break prediction and error correction results.

			Break_Correct	Juncture_Correct	Adjusted_Score
Probabilistic method only			52.17%	81.39%	0.480
IG-Tree only		W = 7	50.58%	81.39%	0.480
		W = 11	51.66%	81.65%	0.487
		W = 15	51.77%	81.71%	0.488
Probabilistic method and post error correction	6:3	before error correction	52.03%	81.29%	0.477
		W = 7	57.34%	83.67%	0.543
		W = 11	59.80%	84.69%	0.572
		W = 15	60.75%	85.06%	0.582
	4:5	before error correction	51.30%	80.85%	0.465
		W = 7	59.04%	84.42%	0.564
		W = 11	61.83%	85.16%	0.585
		W = 15	62.74%	85.57%	0.597

increases from 30% to 50% of the total corpus. This result shows that the proposed architecture can provide improved results even with the phrase break predictor that has poor initial performance.

5 Conclusion

This paper presents a new phrase break prediction architecture that integrates the probabilistic approach with the decision-tree based approach in a synergistic way. Our main contributions include presenting decision tree-based error correction for phrase break prediction. Also, probabilistic phrase break prediction was implemented as an initial annotator of the decision tree-based error correction. The architecture can provide improved results even with the phrase break predictor that has poor initial performance. Moreover, the system can be flexibly tuned to new corpus without massive retraining which is necessary in the probabilistic method. As shown in the result, performance of the hybrid phrase break prediction is determined by how well the error corrector can compensate the deficiencies of the probabilistic phrase break prediction.

The next step will be to analyze the learned decision trees carefully to extract more desirable feature vectors. We are now working on incorporating this phrase break prediction method into the experimental Korean TTS system.

References

J. Allen and S. Hunnicut. 1987. *From Text to Speech: the MITalk System.* Cambridge University Press.

E. Brill. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of the conference on applied natural language processing.*

Jeongwon Cha, Geunbae Lee, and Jong-Hyeok Lee. 1998. Generalized unknown morpheme guessing for hybrid POS tagging of Korean. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 85–93.

Walter Daelemans, Steven Gills, and Gert Durieux. 1994. The acquisition of stress: A data-oriented approach. *Computational Linguitics*, 20(3):421 – 451.

Sangho Lee and Yung-Hwan Oh. 1999. Tree-based modeling of prosodic phrasing and segmental duration for korean tts systems. *Speech Communication*, 28(4):283–300.

Tom M. Mitchell. 1997. *Machine Learning.* McGraw-Hill.

J. R. Quinlan. 1983. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Eric Sanders. 1995. Using probabilistic methods to predict phrase boundaries for a text-to-speech system. Master's thesis, University of Nijmegen.

Paul Taylor and Alan W. Black. 1998. Assigning phrase breaks from part-of-speech sequences. *Computer Speech and Language*, 12(2):99–117.

Paul A. Taylor, I. A. Nairn, A. M. Sutherland, and M. A. Jack. 1991. A real time speech synthesis system. In *Proceedings of the Eurospeech '91.*

Jan P.H. van Santen, Richard W. Sproat, Joseph P. Olive, and Julia Hirschberg. 1997. *Progress in Speech Synthesis.* Springer-Verlag.