Training Prosodic Phrasing Rules for Chinese TTS Systems

Weijun Chen, Fuzong Lin, Jianmin Li, Bo Zhang

Department of Computer Science and Technology
Tsinghua University, Beijing, China
cwj@s1000e.cs.tsinghua.edu.cn

Abstract

This paper describes several experiments designed to train prosodic phrasing models for Chinese TTS systems and to investigate the underlying rules that control Chinese prosody. First, we collected 559 sentences from news programs and built a large corpus for modeling Chinese prosody. Second, we selected 20 features and used classification and regression trees (CART) and transformational rule-based learning (TRBL) techniques to generate phrasing rules automatically. Lastly, we propose a computer aided error-driven method of designing rule templates, and integrate it into the TRBL algorithm. The experimental results show that we achieve a high success rate of 94.5%, and we also get a set of well comprehensible rule templates which may give us insights into the relationship between Chinese syntax and prosody.

1. Introduction

The prediction of prosodic phrase boundaries is an important step for a text-to-speech system. In fact, fluent spoken language is never produced in a smooth, unvarying stream. People tend to group words into phrases and place short pauses between them [1]. Furthermore, variation in phrasing can change the meaning hearers assign to the utterances of a given sentence. Researchers have shown that the relative size and location of prosodic phrase boundaries provides an important cue for resolving syntactic ambiguity [2].

Recently, research on the location of phrase boundaries has been concentrated mainly on acquiring phrasing rules automatically by using some trainable procedures, such as hidden markov models (HMM) [3], neural networks (NN) [4], classification and regression trees (CART) [5][6][7][8] and transformational rule-based learning (TRBL) [9]. These data-driven approaches usually achieve higher success rates than traditional handwritten rules systems, but they also have their own disadvantages. The biggest problem of HMM and NN is thought to be their low comprehensibility in the prediction process. Note that a trained neural network is too difficult for humans to decipher. And although the interpretation of a tree is much more straightforward, simple decision tree predictor can hardly provide us with the explicit rules that reveal the inherent relationship between syntax and prosody. For TRBL, as we will show, the design of a good set of rule templates is the most important.

For Chinese, we know that precise syntactic analysis of the sentence structure is difficult and large amount of computation required. So most of the previous TTS systems just identify the words in the input texts and no more prosodic information is extracted for further processing [10]. In this paper, we describe several experiments designed to train prosodic phrasing models for Chinese TTS systems and to investigate the underlying rules that control Chinese prosody. The aim of this paper is threefold. First, we construct a large speech corpus and annotate the transcribed text for modeling Chinese prosody. Especially we annotate not only part-of-speech information, but also syntactic information. Second, we select linguistic features which seem to affect Chinese prosody and use CART and TRBL techniques to generate phrasing rules automatically. Lastly, we propose a computer aided error- driven method of designing rule templates, and integrate it into the TRBL algorithm. The experimental results show that the TRBL algorithm equipped with our new templates achieves better performance than simple decision tree and constrained TRBL.

2. Corpus for modeling Chinese prosody

For a data-driven method, the scale and quality of the corpus is very important [6][7]. Since there is no suitable corpus available for modeling Chinese prosody, we collected 559 sentences (of approximately 78 min length) from CCTV (China Center Television) news programs and build a corresponding speech corpus uttered by a famous male announcer. We believe that the announcer's professional skills can help to reduce the abnormal phenomena and disagreements occurring in the corpus. Because we found that the transcription of punctuation did not seem to exactly match the original written text, and the punctuation information was also used in some models [6][7], we collected the original script of each piece of news from newspapers and other sources. And then, the annotation of this data for analysis was conducted in the following steps:

First, although a Chinese word is composed of one to several characters, a Chinese sentence is in fact a string of characters without blanks to mark the word boundaries. Therefore the first step is to identify the words in the text corpus. This task was accomplished by using a simple segmentation algorithm and the errors were corrected manually.

Second, we obtained the part-of-speech information of each word via Bai's POS tagger [11], whose output had been modified slightly to adapt to the characteristic of prosody. We elaborately defined 72 POS's which were classified into 20 groups.

Third, the syntactic analysis was done manually. To describe the structure of a sentence, we propose a simplified Chinese grammar. The grammatical functions we considered include subject, predicate, object, attribute, adverbial

modifier and complement. The syntactic phrasal units include noun phrases, prepositional phrases, adjectival phrases, etc.

Lastly, we labeled the speech prosodically by hand, noting location and type of prosodic boundaries. We labeled two levels of boundary, major phrase boundaries and minor phrase boundaries; in the analysis presented below, however, these are collapsed to a single category.

3. Prosodic phrasing using CART

In recent years CART techniques [12] have gained in popularity due to their straightforward interpretation of the resulting tree [5][6][7][8]. By exploring the questions used in the nonterminal nodes, one can evaluate the effectiveness of the feature variables, and get some insight into the given problem. So our first experiment implemented the basic CART techniques and used them to generate the decision tree automatically.

The problem of prosodic phrasing is defined as follows [3]: the input sentence consists of a sequence of words and between each pair of adjacent words is a word juncture $< w_i , w_{i+1} >$, where w_i represents the word to the left of the juncture and w_{i+1} represents the word to the right. There are a number of juncture types and the task of a phrasing model is to assign the most appropriate type to each juncture. The experiments in this paper use two types of juncture, boundary and non-boundary.

We selected 20 feature variables which seem to affect Chinese prosody, and they are categorized into five types: part-of-speech information, syntactic constituency, temporal information, pitch accents and probability information. Most of them are from [5][6]:

- *Pos{1-4}*: A part-of-speech window of four around the juncture, $< w_{i-1} , w_i , w_{i+1} , w_{i+2} >$. The value of each POS variable can be the combination of any POS's of the same group.
- *Sbs*: Smallest constituent dominating both w_i and w_{i+1}.
- *Sll*: Largest constituent dominating w_i, but not w_{i+1}.
- *Srl*: Largest constituent dominating w_{i+1}, but not w_i.
- *Ltw, Ltc*: Total words and Chinese characters in sentence.
- *Lsw, Lsc*: Distance from start to w_i, in words and characters.
- *Lew, Lec*: Distance from w_{i+1} to end, in words and characters.
- *Lllw, Lllc*: The size of *Sll*, in words and characters.
- *Lrlw, Lrlc*: The size of *Srl*, in words and characters.
- *Acc{1-2}*: Whether w_i and w_{i+1} bear a pitch accent or not.
- *Gbipos*: The probability of a boundary occurring within a POS bigram. This probability is calculated from the training data by finding all the sequences of a particular bigram, and dividing the number of boundaries by the total number of occurrences of the bigram.

4. Prosodic phrasing using TRBL

4.1. TRBL

Transformational rule-based learning is an automatically trainable approach which iteratively uses a greedy algorithm to derive an ordered sequence of rules that minimize the overall classification error [13]. The resulting rule sequence is then used to predict new data. TRBL is similar to CART techniques in that they are both corpus-based automatic learning methods and both are designed with greedy algorithms. However, TRBL is thought to be superior to decision trees in two aspects: First, theoretically speaking, the set of classifications that can be provided via decision trees is a proper subset of those that can be provided via sequences of transformational rules. Second, TRBL can help to reveal the underlying rules of a given problem in a clearer and more direct fashion.

A specific application of TRBL depends on three fundamental elements: an initial-state annotator, a set of rule templates, and an objective function for choosing the transformational rules. For example, for the prosodic phrasing, the initial rule could be "assign non-boundary to each word juncture". The templates may take on the form "if *Lew* (the distance from w_{i+1} to end in words) is less than *N* words, then change the type of this juncture to non-boundary". And the objective function is likely to be minimum classification error.

The learning process of TRBL can be summarized as follows [9]:

First, apply some base rules to the unannotated input text to create an initial state.

Second, for all possible rule templates, all possible features and all possible feature values, instantiate a rule from a template, and calculate its score (reduction of error, for example) by experimentally applying the rule to the data.

Third, find the rule whose application results in the best score according to the objective function being used.

Lastly, if the best score exceeds a threshold, update the state of the data according to this rule and go to the second step. Otherwise, stop the algorithm.

4.2. Method of designing rule templates

As our experimental results will show, the design of the rule templates is very important. A good set of templates can significantly improve the performance, and contrarily, although TRBL is thought to be more powerful than decision trees in theory, when using the rule templates that are equivalent to the types of questions that can be asked in the decision tree, we find that CART achieves better performance than TRBL. Here we propose a computer aided error-driven method of designing rule templates semiautomatically, and the learning process of TRBL is expanded as follows:

1. Apply some base rules to the unannotated text to create an initial state. And currently the template list is empty.
2. Analyze the error samples and propose a possible rule template. If fail to do so, stop.
3. For the new template, and for all possible features and feature values, instantiate a rule from the template and

calculate its score by experimentally applying the rule to the data.

4. Find the rule whose application results in the best score, and examine the errors it caused. Then judge whether it is necessary to refine the template, if necessary, refine it and go to step (3). Otherwise, add this template to the template list, and change the data to the initial state.
5. For all templates in the template list, all possible features, and all possible feature values, instantiate a rule from a template, and calculate its score.
6. Find the best rule, and if its score exceeds a threshold, update the data according to the rule and go to step (5). Otherwise, go to step (2).

4.3. Rule templates for Chinese prosodic phrasing

Using the method described above, we spent about two weeks designing the rule templates for Chinese prosodic phrasing. There are a total of 30 templates and the first 13 are listed below (Note that our initial rule is "assign non-boundary to each word juncture"):

1. If *Sll* is a subject, *Lllc* > *M*, and *Lrlc* > *N*, then change the type of this juncture to "boundary".
2. If *Sll* is a predicate, and w_i or w_{i+1} bears a pitch accent, then change the juncture type to "boundary".
3. If *Sll* is an adverbial modifier and a prepositional phrase, *Lllc* > *M*, and *Lrlc* > *N*, then change the juncture type to "boundary".
4. If w_{i+1} is a comma, then change the juncture type to "boundary".
5. If w_{i+1} is the word "和" (and), *Lllc* > *M*, and *Lrlc* > *N*, then change the juncture type to "boundary".
6. If *Sll* is an object, and w_{i+1} is a verb, and *Lllc* > *N*, then change the juncture type to "boundary".
7. If w_i is the word "的" ('s), and w_{i+1} bears an accent, then change the juncture type to "boundary".
8. If *Sll* is an adverbial modifier and not a prepositional phrase, and *Lllc* > *N*, then change the juncture type to "boundary".
9. If *Sll* is an attribute and *Lllc* > N, and w_{i+1} bears an accent, then change the juncture type to "boundary".
10. If *Sll* is an attribute, the POS of w_i is "nP" (a person's title), and the POS of w_{i+1} is "nN" (a person's name), then change the juncture type to "boundary".
11. If w_i is the word "的" ('s), *Lllc* > *M* and *Lrlc* > *N*, then change the juncture type to "boundary".
12. If the juncture is a predicate-subject boundary and *Lllc* > *N*, then change the juncture type to "boundary".
13. If *Sll* is a noun phrase, *Srl* is a prepositional phrase, and *Lllc* > *N*, then change the juncture type to "boundary".

5. Results

5.1. Results

To evaluate the performance of our method, we divided the corpus into training data (371 sentences, 8170 word junctures) and test data (188 sentences, 4084 word junctures). We did not consider the boundaries between sentences because virtually there is always a boundary at the end of a sentence. The experimental results are shown in Table 1. In order to provide a baseline, accuracy using the initial rule that assigns "non-boundary" to each word juncture is also shown. The constrained TRBL experiment is the same as [9], which used the rule templates that were equivalent to the types of questions that could be asked in the decision tree.

We got the same result as [9] that the decision tree gives better performance than the constrained TRBL. But we think this is because most of the underlying rules that reveal the relationship between syntax and prosody are *N*-feature (*N* > 2) combination ("and") rules (just as our templates have shown), and CART technique provides a mechanism to produce such types of rules (the path from the root node to a leaf node is such a combination) whereas TRBL doesn't. And the limitation of computational costs does not allow arbitrary combination of features, so the key problem of TRBL is to design a good set of rule templates.

Table 1: Performance of different methods.

Algorithm	Accuracy
All no boundary	77.6%
CART	91.4%
Constrained TRBL	89.5%
TRBL + new templates	94.5%

The confusion matrix for the TRBL experiment with new templates is given in Table 2. We analyzed the error samples and found that most of the "no boundary" errors are the exceptions of the resulting rules, and the "boundary" errors can be classified into three types: exceptions of the rules, instances of trivial rules whose templates were not included, and unsure errors.

Table 2: Confusion matrix for TRBL + new templates.

Truth	Predicted		
	No boundary	Boundary	Per cent correct
No boundary	3102	52	98.4%
Boundary	173	757	81.4%

5.2. Tradeoff between accuracy and template number

Theoretically speaking, we can continue to improve the prediction accuracy by adding more and more templates. But more templates mean more labor and may take the risk of overfitting. So there is a tradeoff between the prediction accuracy and template number. In fact, TRBL has arranges the order of the rules according to their effectiveness in the learning process. So we can just choose the first *N* rules as the resulting rule sequence if they have already satisfied our

accuracy requirement. In our experiment, the relationship between accuracy and template number is shown in Figure 1.

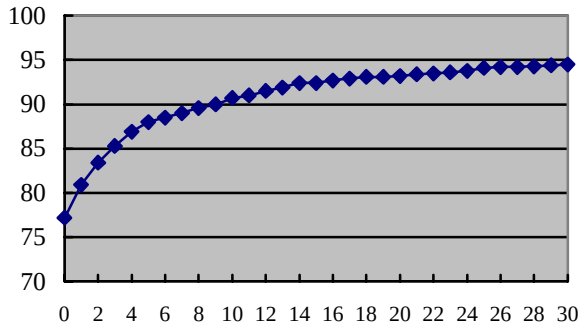

Figure 1: This plot shows the relationship between prediction accuracy and template number. The x-axis gives the number of template and the y-axis gives the accuracy.

5.3. Contribution of accent to prosodic boundary

It was reported in [9] that accent prediction benefits from phrase structure, but not vice versa. However, in our experiments, we find that accent location is also used in phrase boundary prediction. To assess the impact of accent, we retrained the rules without using any accent information. The results are presented in Table 3.

Table 3: Performance of different methods without using accent information.

Algorithm	Accuracy
All no boundary	77.6%
CART	88.8%
Constrained TRBL	87.7%
TRBL + new templates	90.9%

Table 3 shows that not using accent location will degrade the performance significantly, which is quite different from [9]. But we think this may be due to the difference in the language. In our speech corpus, we observed that the word junctures before and after a pitch accent are more likely to be prosodic boundaries.

6. Conclusions

We summarize our conclusions as follows: (1) Both CART and TRBL are effective techniques in modeling prosodic phrasing for Chinese TTS systems. And although TRBL is thought to be more powerful than CART in theory [13], the decision tree gives better performance than the constrained TRBL. But we think this may depend on particular problems. In the prediction of pitch accent locations [9], for example, contrary result was reported. (2) The new defined rule templates improved the performance remarkably, and they can also help us better understand the relationship between Chinese syntax and prosody. (3) Theoretically we can continue to improve the prediction accuracy by adding more and more templates, but there is always a tradeoff. (4) For Chinese, phrase boundary prediction benefits from accent location significantly.

Using our computer aided error-driven method, it is possible to produce a good set of rule templates easily and quickly. However, when we examined the prediction errors, we found that some trivial rule templates had been ignored by us. For ongoing research to further improve the performance, we are experimenting to produce the rule templates automatically by using some machine learning algorithms.

Acknowledgments: This research was supported by the Natural Science Foundation of China (69823001), and Doctoral Program Foundation (98000335).

7. References

[1] Wightman, C. W., Ostendorf, M. and Price, P. J. “Segmental durations in the vicinity of prosodic phrase boundaries”. *J. Acoust. Soc. Amer.*, 91(3):1707-1717, 1992.

[2] Ostendorf, M. and Wightman, C. W. “Parse scoring with prosodic information: an analysis/synthesis approach”. *Computer Speech and Language* , 7:193-210, 1993.

[3] Taylor, P. and Black, A. W. “Assigning phrase breaks from part-of-speech sequences”. *Computer Speech and Language*, 12:99-117, 1998.

[4] Muller, A. F., Zimmermann, H. G. and Neuneier, R. “Robust Generation of Symbolic Prosody by a Neural Classifier Based on Autoassociators”. In *Proc. ICASSP*, 2000. pp. 1285-1288.

[5] Wang, M. Q. and Hirschberg, J. “Automatic classification of intonational phrase boundaries”. *Computer Speech and Language* , 6:175-196, 1992.

[6] Hirschberg, J. and Prieto, P. “Training intonational phrasing rules automatically for English and Spanish text-to-speech”. *Speech Communication*, 18:281-290, 1996.

[7] Lee, S. and Oh, Y.H. “Tree-based modeling of prosodic phrasing and segmental duration for Korean TTS systems”. *Speech Communication* . 28:283-300, 1999.

[8] Koehn, P., Abney, S., Hirschberg, J. and Collins, M. “Improving intonational phrasing with syntactic information”. In *Proc. ICASSP*, 2000. pp.1289-1290.

[9] Fordyce, C. S. and Ostendorf, M. “Prosody Prediction for Speech Synthesis Using Transformational Rule-Based Learning”. In *Proc. ICSLP*, 1998. pp. 682-685.

[10] Chou, F. C., Tseng, C. Y., Chen, K. J. and Lee, L. S. “A Chinese Text-to-Speech System Based on Part-Of-Speech Analysis, Prosodic Modeling and Non-uniform Units”. In *Proc. ICASSP*, 1997. pp. 923-926.

[11] Bai S. H. “The Study and Realization of Statistics Based Approach to Tagging Chinese Corpus”. Master thesis, Tsinghua University, 1992. (In Chinese)

[12] Breiman, L., Friedman, J., Olshen, R and Stone, C. *Classification and Regression Trees.* Belmont, CA: Wadsworth, 1984.

[13] Brill, E. “Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging”. Computational Linguistics. 21(4): 543-565, 1995.