Automatic Feature Template Generation for Prosodic Phrasing

Fangzhou Liu
College of Mathematics and Computer Science, Hunan Normal University, Changsha, China
ark.new@gmail.com

You Zhou
Department of Applied Mathematics, Hunan University of Finance and Economics, Changsha, China
peipei.new@gmail.com

***Abstract*—Prosodic phrase prediction is important for both the naturalness and intelligibility of Text-to-Speech (TTS) systems. To automatically generate feature templates of prosodic phrasing models, this paper proposes a hybrid approach which converts the rules generated by classification and regression tree (CART) into templates of transformation-based learning (TBL), and designs a hierarchical clustering based feature combination algorithm for maximum entropy (ME) model. While minimizing human supervision, TBL templates automatically generated by CART can provide good alternatives or beneficial supplement to manually summarized templates, and ME templates automatically generated by the proposed feature combination algorithm not only make an improvement of 3.1% on F-measure over manual templates, but also reduce the size of ME model by up to 79.0%.**

***Index Terms*—prosodic phrase prediction, feature template generation, keyword selection, classification and regression tree (CART), transformation-based learning (TBL), maximum entropy (ME).**

I. INTRODUCTION

In spoken language, prosodic phrases (hereinafter referred to as "PP") are separated by breaks in the form of pauses. PP breaks divide utterances into information chunks to make understanding faster and easier. Variation in prosodic phrasing can change the meaning got by listeners from the same sentence. Situations where PP breaks are lost when necessary or added in wrong places make the synthetic speech sound unnatural and boring. Therefore, PP break prediction is of great importance for both the intelligibility and naturalness of Text-to-Speech (TTS) systems.

The prediction of PP breaks is usually regarded as a classification problem, where a classifier is used to make decision whether each word boundary is a PP break or not. Various data-driven approaches, such as Markov model [1], classification and regression tree (CART) [2], transformation-based learning (TBL) [3], maximum entropy (ME) model [4] etc, have been investigated to predict PP breaks. Although most of them achieve state-of-the-art performance, these methods still have their own shortcomings. Markov model can only use a few discrete features like part-of-speech (POS), and is based on the strict Markov assumption, which has a limited validity. CART recursively splits the data, and hence suffers from unreliable counts due to data fragmentation. For the task of PP break prediction, both TBL and ME model outperform CART [3] [4], but manually summarizing templates for them is time-consuming and laborious, and because of the limitation of human knowledge and ability, it is difficult to cover enough and accurate phrasing rules with manual templates.

This paper also uses TBL and ME model, which have been proved outstanding in previous work, to predict PP breaks from unrestricted Chinese text. Furthermore, two kinds of objective methods, the CART rule conversion approach and the feature combination algorithm, are proposed to automatically generate feature templates of prosodic phrasing models. The former converts the rules generated by CART into TBL templates. Experimental results show that CART templates can provide good alternatives or beneficial supplement to manual templates. The latter chooses the optimal couples of features to combine into templates. While reducing human supervision during ME training, the feature combination algorithm not only obviously improves the accuracy of PP break prediction, but also significantly reduces the number of templates, thus greatly minifying the size of ME model.

II. FRAMEWORKS OF CLASSIFIERS

A. Transformation-Based Learning

Transformation-based learning (TBL), first proposed by E. Bill [5], is a rule-based machine learning algorithm. It has been applied since to various natural language processing (NLP) tasks, including part of speech tagging [5], noun phrase chunking [6], parsing [7] etc, often

Supported by the Science and Technology Planning Project of Hunan Province (No. 2010FJ4131) and the Scientific Research Foundation of Hunan Provincial Education Department (No. 10C0955).
Corresponding author: Fangzhou Liu
Email: ark.new@gmail.com

doi:10.4304/jsw.7.4.779-785

achieving state-of-the-art performance with a small and easily-understandable list of rules.

The central idea of TBL is to learn an ordered list of transformation rules which progressively improve the current state of the training set. Fig. 1 illustrates the learning process. At first, an initial assignment is made based on simple statistics, and then transformation rules are greedily learned to correct the mistakes, until no net improvement can be made upon the training set. During the evaluation phase, the evaluation set is initialized by the same initial-state annotator. Each rule is then applied, in the order it was learned, to the evaluation set. The final classification is the one attained when all rules have been applied.

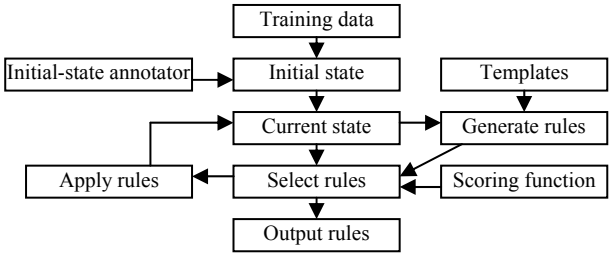

Figure 1. Training process of TBL.

A template of TBL consists of several features, such as "P_{-2}&P_{-1}:X→Y", where P denotes the POS, X and Y denote the break label (PP break or not) before and after the transformation respectively, and the subscript number denotes the offset from the current word boundary. Templates determine the predicates of rules, and have the greatest influence on the performance of TBL.

B. Maximum Entropy Model

Maximum entropy (ME) model is a powerful statistical classification model, which has been successfully applied to a variety of tasks, including word segmentation [8], POS tagging [9], word sense disambiguation [10] etc.

The central idea of ME model is to seek the probability distribution with the maximum entropy, over the set of the ones subject to certain constraints. Such constraints force the model to match its feature expectations with those observed in the training set. A feature of ME model is a binary-value function $f_i(x, y)$ as follows:

$$f_i(x, y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are both true} \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

In the scenario of PP break prediction, x denotes the contextual information, and y denotes the break label. Given an event (x, y), the conditional probability $p(y \mid x)$ can be computed as:

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} . \quad (2)$$

Where α_i is the weight of feature f_i, which can be estimated by GIS or IIS [11] algorithm, and $Z(x)$ is the normalization factor:

$$Z(x) = \sum_{y} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} . \quad (3)$$

A trained ME model is composed of its features and their weights, so the performance of ME model depends largely on the effectiveness of the features, which is extracted from the training corpus according to the feature templates. Therefore, feature templates determine the value space of features, and have the greatest influence on the behavior of ME model.

III. AUTOMATIC TEMPLATE GENERATION

The common method of feature template acquisition is to select the basic features which are discriminative for classification, and then sum up the patterns of feature co-occurrence by observing the annotated corpus. However, that process is time-consuming and laborious, and because of the limitation of human knowledge and ability, it is difficult to cover enough and accurate phrasing rules with manual templates. To overcome the shortcomings of manual template acquisition, we propose two objective approaches of automatic template generation.

A. Converting CART Rules into Templates

As is well known, each non-leaf node of CART specifies a test about some discriminative features for classification, and the path from the root to a leaf node gives a decision rule which is composed of those features the path passes. The proposed approach attempts to convert the rules got from leaf nodes of CART into templates. As shown in Fig. 2, template "L_{-1}&P_1" can be derived from the leaf node A, and template "L_{-1}&P_1&P_2" can be derived from the leaf node B. For abbreviation, the templates automatically generated by CART are referred to as the "CART templates" in the remainder of the paper.

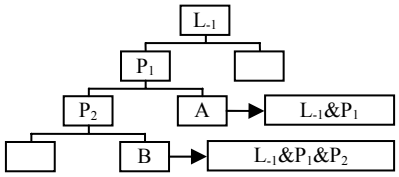

Figure 2. Converting CART rules into TBL templates.

However, CART templates have a flaw that can not include the specific word feature, which can provide important cues for identifying PP breaks [4]. It is because while splitting the data set in training process, CART needs to try various divisions of feature values to select the optimal division. Therefore, the word feature whose number of values is up to ten thousand can not be selected as the branch node of CART, due to the limitation of time and memory. In order to make CART can use the word feature, we try to reduce the number of word feature values to a small number of keywords. Keyword selection methods will be described in Section IV.C.

Moreover, for generating TBL templates with CART rules, the different characteristics between these two algorithms should be considered. Rules learned by TBL are used to correct the errors made by the initial-state

annotator, whereas rules generated by CART are aimed at all the training data, including both the correct samples correctly initialized by the TBL initial-state annotator and the error samples the initial-state annotator incorrectly annotates. So TBL templates should meet two requirements. First, they should be able to correct errors. Second, they should not transform correct samples into the wrong ones. Since the majority of training set are correct samples, the CART rules based on all the training data mainly predict the PP breaks of correct samples, so templates converted from these rules (hereinafter referred to as the "training data templates") may be weak in error correction. The CART templates only based on the error data in the training data (hereinafter referred to as the "error data templates") are just the opposite. They are aimed at the error samples, and thus meet the first requirement very well, but are easy to make wrong transformations, due to lack of the supervision of correct samples. Therefore, both the training data templates and error data templates should be taken into consideration.

B. Combining Features into Templates

Leaf nodes of CART have dominating categories, so rules got from them make classification independently, and hence CART templates derived from these rules are independent of each other or even mutually exclusive. However, ME model integrates all the features into an exponential framework. If the feature templates of ME model are mutually exclusive, then its advantage that all the features contribute to classification together by their weight coefficients will not come out to play. Thus, it can be seen that CART templates may not be suitable for ME model.

Based on the idea of hierarchical clustering, we design another iterative algorithm for automatic feature template generation. It uses the F-score of prosodic phrasing model to measure the effectiveness of a feature template. At each iteration, the pair of features whose combination most improves the F-score are combined into a feature template. This iteration is repeated until the combination of any pair of features in the current feature set can not achieve a higher F-score. The specific steps of our iterative algorithm are described as follows:

a) Set the initial feature set $FEASET_0$ to be the basic feature set, the current feature set $FEASET_{cur}$ to be $FEASET_0$, the initial template set $TPLSET_0$ to be empty, its F-score F_0 to be 0, the current template set $TPLSET_{cur}$ to be $TPLSET_0$, and its F-score F_{cur} to be F_0.

b) For each pair of features FEA_i and FEA_j in $FEASET_{cur}$, try to merge them into a new template $TPL_{i\&j}$, temporarily add $TPL_{i\&j}$ into $TPLSET_{cur}$ to get the temporary template set $TPLSET_{tmp}$, and calculate its F-score $F_{i\&j}$. Then find out the maximum $F_{i\&j}$ (denoted by $F_{i\&j\ max}$) and the corresponding template $TPL_{i\&j\ max}$.

c) If $F_{i\&j\ max}$ is lower than F_{cur}, go to step *d*; otherwise, make F_{cur} equal to $F_{i\&j\ max}$, formally add the new template $TPL_{i\&j\ max}$ into $TPLSET_{cur}$, add $TPL_{i\&j\ max}$ into $FEASET_{cur}$ as a new feature too, and then return to step *b*.

d) The iteration ends. Output the current template set $TPLSET_{cur}$.

IV. Feature Selection

A. Feature Set

According to Liu's statistics [12], most Chinese prosodic phrases are of length of 3 to 6 syllables, that is, about 2 lexicon words long. Therefore, 4 kinds of contextual information in a window of -2~+2 words (the distance features are beyond the scope of this window) are employed in this work. All the features are listed in Table I, where W denotes the lexicon word, P, S and L denote the POS, semantic class and length in syllables of the lexicon word respectively, DB and DE indicate the distance in syllables to the beginning and ending of the sentence respectively, and the subscript number denotes the offset from the current word boundary. Note that the semantic classes of notional words are obtained through looking up in a semantic dictionary according to their POS. In case of semantic ambiguity, only the most frequent sense is adopted.

TABLE I. Feature Set

Feature type	Value range
W_{-2}, W_{-1}, W_1, W_2	About 25,000 entries
P_{-2}, P_{-1}, P_1, P_2	39 categories
S_{-2}, S_{-1}, S_1, S_2	67 classes
L_{-2}, L_{-1}, L_1, L_2	1~8 syllables
DB, DE	1~32 syllables

B. Manual Template Set

Based on empirical knowledge, the basic features listed in Table I are combined into 66 templates. Some features, like W, P, S, DB and DE, are also used as a template alone. Table II lists all the manual templates.

TABLE II. Manual Template Set

W_{-2}, W_{-1}, W_1, W_2, $W_{-2}W_{-1}$, $W_{-1}W_1$, W_1W_2, $W_{-2}W_{-1}W_1$, $W_{-1}W_1W_2$
P_{-2}, P_{-1}, P_1, P_2, $P_{-2}P_{-1}$, $P_{-1}P_1$, P_1P_2, $P_{-2}P_{-1}P_1$, $P_{-1}P_1P_2$
S_{-2}, S_{-1}, S_1, S_2, $S_{-2}S_{-1}$, $S_{-1}S_1$, S_1S_2, $S_{-2}S_{-1}S_1$, $S_{-1}S_1S_2$
$W_{-2}P_{-1}$, $W_{-1}P_1$, W_1P_2, $P_{-2}W_{-1}$, $P_{-1}W_1$, P_1W_2
$W_{-2}P_{-1}P_1$, $W_{-1}P_1P_2$, $P_{-2}W_{-1}P_1$, $P_{-1}W_1P_2$, $P_{-2}P_{-1}W_1$, $P_{-1}P_1W_2$
$S_{-2}P_{-1}$, $S_{-1}P_1$, S_1P_2, $P_{-2}S_{-1}$, $P_{-1}S_1$, P_1S_2
$S_{-2}P_{-1}P_1$, $S_{-1}P_1P_2$, $P_{-2}S_{-1}P_1$, $P_{-1}S_1P_2$, $P_{-2}P_{-1}S_1$, $P_{-1}P_1S_2$
$L_{-2}L_{-1}$, $L_{-1}L_1$, L_1L_2, $L_{-2}L_{-1}L_1$, $L_{-1}L_1L_2$
$P_{-2}L_{-2}$, $P_{-1}L_{-1}$, P_1L_1, P_2L_2, $P_{-2}L_{-2}P_{-1}L_{-1}$, $P_{-1}L_{-1}P_1L_1$, $P_1L_1P_2L_2$
DB, DE, DB&DE

After extracting feature instances from the training set according to the feature templates, it is also needed to filter the features in order to remove noise. There are usually two algorithms to select features, CCFS (Count Cutoff Feature Selection) and IFS (Incremental Feature Selection) [13]. In CCFS algorithm, features which occur very sparsely in the training set are considered to be statistically unreliable, and should be excluded out of the feature set. A threshold (cutoff value) on frequency count is set to omit those features. IFS is an iterative algorithm which measures the effectiveness of a feature based on its contribution to the likelihood. At each iteration, the

feature which most increases the likelihood of the training data is added into the feature set. Adwait [13] pointed out that, IFS algorithm had heavy computation complexity, and took much long time to train the model, while did not always outperform CCFS algorithm. In this paper, CCFS algorithm is applied to feature selection.

C. Keyword Selection

A word is considered as a keyword when its presence or absence gives more information than others. There are a variety of keyword selection techniques to measure the amount of this information in various domains. For instance, log-likelihood ratio is used to disambiguate word sense [14]. Mutual information and information gain, which are both well known as information measures, are used in text classification [15]. Cross entropy [16] is similar to information gain, with the difference that the former ignores the absence of feature. Odds ratio is commonly used in information retrieval [16]. All 5 keyword scoring measures mentioned above are listed in Table III, where P denotes the probability. For example, $P(W)$ is the probability that word W occurs. $\overline{W}$ means word W does not occur. B denotes the break label, B_0 means the target word boundary is not a PP break, and B_1 is the opposite. Since there are only two types of break labels, $\overline{B_0} = B_1$ and $\overline{B_1} = B_0$.

TABLE III. KEYWORD SCORING MEASURES

$LogLikelihoodRaito(W) = P(W)\sum_{i=0}^{1} P(B_i)\left\lvert\log(\frac{P(B_i \mid W)}{P(B_i \mid \overline{W})})\right\rvert$
$MutualInformation(W) = \sum_{i=0}^{1} P(B_i)\log\frac{P(W \mid B_i)}{P(W)}$
$InformationGain(W) = P(W)\sum_{i=0}^{1} P(B_i \mid W)\log\frac{P(B_i \mid W)}{P(B_i)} + P(\overline{W})\sum_{i=0}^{1} P(B_i \mid \overline{W})\log\frac{P(B_i \mid \overline{W})}{P(B_i)}$
$CrossEntropy(W) = P(W)\sum_{i=0}^{1} P(B_i \mid W)\log\frac{P(B_i \mid W)}{P(B_i)}$
$OddsRatio(W) = P(W)\sum_{i=0}^{1} P(B_i)\left\lvert\log\frac{P(W \mid B_i)(1-P(W \mid \overline{B_i}))}{(1-P(W \mid B_i))P(W \mid \overline{B_i})}\right\rvert$

V. CORPUS ANNOTATION

A large corpus of 14,314 sentences and around 196,600 syllables was collected from the "People's Daily", which had been automatically preprocessed by the front end of our Mandarin TTS system, including word segmentation and POS tagging. The F-score of word segmentation achieved 96.7%, and the precision of POS tagging was 92.3%.

By reading the text transcriptions, PP breaks were annotated manually by two experienced annotators. To check the consistency of annotation between the two annotators, an exploratory experiment was carried out. The two annotators were first trained on the same 100 sentences. At this stage, they were required to discuss the annotation criteria so that they could achieve agreement on most of the annotations in the 100 sentences. Then, they were asked to annotate a small subset of the corpus, which included 1,000 sentences. Taking one of the two annotators as the reference, the other one as a classifier, the F-score of manual annotation is 85.2%, which can be considered as the upper limit of the automatic allocation of PP breaks.

This annotated corpus is divided into the training set, development set and test set according to an 8:1:1 ratio. The score threshold of TBL, the iteration times of ME model and the cutoff value are set on the development set. CART pruning and keyword selection are implemented on the development set too.

VI. EVALUATION AND DISCUSSION

A. Evaluation Criteria

The performance of proposed approaches is measured by F-score, which is the harmonic mean of precision and recall.

$$F\text{-}score = \frac{P \times R}{(1-\alpha) \times P + \alpha \times R}. \quad (3)$$

Here, P and R denote precision and recall respectively. α is a factor determining the weighting of precision and recall. The value of α = 0.5 has been used in the current evaluation for equal weighting of precision and recall.

Note that the evaluation process only considers PP breaks within a sentence, irrespective of the ones at the end of the sentence.

B. Experiments of Keyword Selection

Selection of keyword scoring measures: To compare the performance of keyword selection methods, 50 keywords with the highest score are selected as the word feature values of CART from the training set, and tested on the development set. CART models are trained with the basic features listed in Table I. Table IV shows the results.

TABLE IV. COMPARISON OF KEYWORD SELECTION MEASURES

Method	F-score of CART
Log-likelihood ratio	74.8%
Mutual information	71.0%
Information gain	73.6%
Cross entropy	74.1%
Odds ratio	74.6%

Our Experimental results are generally consistent with that of Yang [15] and Mladenic [16]. Mutual information has inferior performance compared with the other methods due to its bias favoring rare terms. When a word occurs rarely in the corpus, it may only occasionally appear in the context of a particular type of break labels, leading to a very large mutual information value with this kind of labels. Hence a number of low-frequency words are chosen to be keywords by the mutual information measure, but they do not actually have the statistical significance. The main reason that information gain performs worse than cross entropy is that CART can hardly use the templates of word absence whose information accounts for a large proportion of information gain, so cross entropy which only uses the information of word presence is more suitable for CART

than information gain. The performance of odds ratio is slightly worse than that of log-likelihood ratio which is the simplest but also the most effective approach.

Keyword number setting: Table V compares the performance of CART models with different number of keywords selected by log-likelihood ratio. The keyword number 0 means the CART model without word features, whose performance can be taken as the baseline.

TABLE V. KEYWORD NUMBER SETTING

Keyword number	F-score of CART
0	71.2%
50	74.8%
100	72.0%
150	70.3%

As can be seen from Table V, the CART model with 50 keywords makes an improvement of 5.1% on F-score over the one without word features, which proves the effectiveness of the specific word feature once again. However, with the increase in the number of keywords, the performance of CART decreases rapidly. When the number of keywords reaches 150, its F-score is even lower than not using word feature. The reason may be the increase of number of keywords makes word features more and more sparse, coupled with the data fragmentation problem in CART training, resulting in the performance of CART decreases significantly. Although fewer keywords seem to lead to better performance of CART, the purpose of keyword selection is to make CART can use the word feature. If keywords are too few, they will not have a general representation of the word feature, so we set the number of keywords 50.

C. Cutoff Value Setting

In order to determine the optimal cutoff value, we compare the performance of ME models with various cutoff values on the development set. ME models are trained with the manual templates listed in Table II. Table VI shows the results. The best cutoff value is 2, that is, the features appear less than or equal to 2 times in the training set are thrown out.

TABLE VI. CUTOFF VALUE SETTING

Cutoff Value	F-score of ME
0	74.6%
1	74.5%
2	74.8%
3	74.6%
4	74.7%
5	74.5%

It is observed that most of the discarded low-frequency features are word features. Although these features have a strong ability to distinguish between PP break labels, but they are not statistically stable enough, and easily lead to over-fitting, so it is helpful to delete them for improving the overall quality of feature set. However, if the cutoff value is set too high, a lot of useful information will be lost, thus reducing the performance of ME model. So it is necessary to choose a suitable cutoff value.

D. Experiments of TBL Template Generation

As analyzed in Section III.A, CART templates for TBL include the error data templates and the train data templates. The former is good at error correction, but easy to make wrong transformations, while the latter is just the opposite. Their performance is compared in Table VII, where the basic features listed in Table I are referred to as the atom templates, whose performance can be taken as the baseline. All the CART templates make significant improvement on F-score over the atom templates. The error data templates have inferior performance compared with the training data templates. Combining the two achieves the best F-score. That is because it combines the advantages of both two kinds of CART templates.

TABLE VII. COMPARISON OF CART TEMPLATES IN TBL

Template type	F-score of TBL	Relative improvement
Atom templates	69.6%	-
Training data templates	77.7%	11.6%
Error data templates	76.4%	9.8%
Training data templates & error data templates	78.4%	12.6%

Table VIII compares the performance of manual templates in TBL with that of CART templates (including both the training data templates and the error data templates). The F-score of CART templates almost reaches the level of manual templates, which proves the effectiveness of our CART rules conversion method. Combining the two achieves the best performance. Therefore, CART templates can be used as good substitutes for manual templates, and are still able to provide beneficial supplement for manual templates even if manual templates have been summarized.

TABLE VIII. COMPARISON BETWEEN CART TEMPLATES AND MANUAL TEMPLATES IN TBL

Template type	F-score of TBL	Relative improvement
Atom templates	69.6%	-
CART templates	78.4%	12.6%
Manual templates	78.7%	13.1%
CART templates & manual templates	79.5%	14.2%

E. Experiments of ME Template Generation

Given the good performance of CART templates in TBL, we try to apply them to ME model. Since ME model does not need to correct error samples, the CART templates used by ME model are trained from all the training data, that is, the training data templates. Table IX compares the performance of training data templates in ME model with that of manual templates.

TABLE IX. COMPARISON BETWEEN CART TEMPLATES AND MANUAL TEMPLATES IN ME MODEL

Template type	F-score of TBL	Relative improvement
Atom templates	75.4%	-
CART templates	76.3%	1.2%
Manual templates	78.2%	3.7%

As can be seen from Table IX, the performance of CART templates in ME model is not very good. Its F-score is only 1.2% higher than that of the atom templates, and almost 2.4% lower than that of manual templates. The effect of CART templates in ME model is not as obvious as in TBL for two reasons. First, all the features of ME model are integrated within the exponential framework and contribute to the classification together by their weight coefficients. However, rules learned by CART are independent of each other or even mutually exclusive. Like CART, TBL is also a rule learning algorithm, and E. Bill [5] has theoretically proved that the decision tree algorithm can be regarded as a special case of TBL. Therefore, CART templates are suitable for TBL, but not for ME model. Second, most of CART templates are very complicated, and even consist of more than ten basic features sometimes. The feature instances extracted from these complicated CART templates will be discarded in the feature cutoff due to their low frequency of occurrence. In other words, most of CART templates don't actually generate valuable features of ME model.

The performance of templates automatically generated by the feature combination algorithm (hereinafter referred to as the "feature combination templates") and that of manual templates are compared in Fig. 3, where the horizontal axis gives the number of feature combination templates, which increases from 1 to 12, and the vertical axis gives the F-score of the ME models trained with the feature combination templates. The dashed line indicates the F-score of manual templates, which is 78.2%.

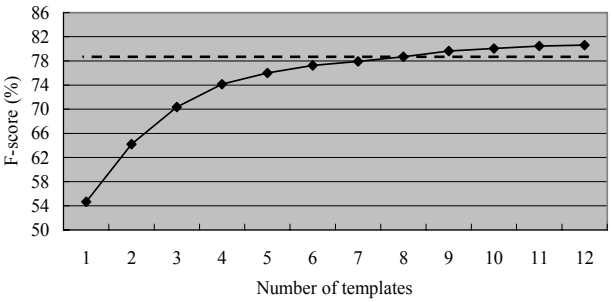

Figure 3. Performance of feature combination templates.

Along with the increase of the feature combination templates, the F-score of ME model grows by and large. When the number of feature combination templates is increased to 8, its F-score reaches the level of 66 manual templates. When the number of feature combination templates increases to 12, its F-score reaches 80.6%, 3.1% higher than that of 66 manual templates. It is thus clear that the feature combination algorithm is very effective. While obviously improving the accuracy of PP break prediction, it greatly reduces the number of feature templates.

However, the feature combination algorithm has the disadvantage that its convergence time is too long. At each iteration, it tries to merge every pair of features in order to find the new template which most improves the F-score. Therefore, if there are n features in the current feature set, to find the best new template of current iteration will require C_n^2 times feature combinations and C_n^2 times ME model training, which will take an intolerably long training time. The basic feature set listed in Table I contains a total of 18 features. To automatically generate 12 feature templates, for example, the times of ME model training reaches $C_{18}^2 + C_{19}^2 + \cdots + C_{29}^2 + C_{30}^2 =$ 3679 times. In order to shorten the convergence time of the algorithm, we replace the convergence condition "$F_{i\&j\ max}$ lower than F_{cur}" with "$F_{i\&j\ max} - F_{cur} \leq \Delta F$", and take the threshold ΔF as 0.1%. Even so, the iteration ends after almost 3 days when 12 feature combination templates are generated.

The good news is that the rewards of long training are considerable. Compared with 66 manual templates, 12 feature combination templates not only obviously improve the performance of ME model, but also reduce the number of ME features from 116,841 to 24,996, and decrease the size of ME model by up to 79.0%, from 1.02M to 219K, thereby significantly minifying the space and memory consumption of the PP break prediction module in TTS systems.

F. Comparison of Classifiers

The performance of CART, TBL and ME model are compared in Table X, where all the 3 algorithms don't use manual templates. CART models are trained with the basic features listed in Table I, TBL rules are learned from the CART templates, and ME models are trained with the feature combination templates. ME model achieves the best performance which is approaching the consistency of manual annotation, TBL ranks second, and CART is the worst.

TABLE X. COMPARISON OF CLASSIFIERS

Classifier	F-score
CART	74.1%
TBL	78.4%
ME	80.6%
Manual annotation	85.2%

CART has inferior performance compared with TBL due to its serious fragmentation problem. Recursively splitting the data set with sparse attributes in CART training will generate too small sub-sample sets, thus easily leading to model over-fitting. The reason ME performs better than TBL is because ME model has a good ability of quantitative description. By the weight coefficient, ME model can accurately describe the contribution of each feature to the classification. Thus all the features are integrated effectively within the same framework, rather than only several features being simply merged together into a rule just as in TBL.

VII. CONCLUSION

In this paper, transformation-based learning (TBL) and maximum entropy (ME) model are investigated to predict prosodic phrase (PP) breaks. In order to reduce human supervision during the training process, we propose two approaches of automatic feature templates generation. The CART rule conversion approach converts the rules

got from leaf nodes of CART into feature templates. The excellent performance of CART templates in TBL indicates that they can provide good alternatives or beneficial supplement to manual templates. Furthermore, comparative experiments on TBL template generation demonstrate that it is very important to consider the error-driven characteristic of TBL. The feature combination algorithm combines the optimal pairs of features into feature templates iteratively. While obviously improving the accuracy of PP break prediction, it greatly reduces the size of ME model, thus effectively optimizing the model size and memory consumption of prosodic phrasing.

Compared with inter-annotator agreement, the F-score of ME model seems to be approaching the up limit of PP break prediction based on our corpus. To further increase the prediction accuracy needs to improve the consistency of manual annotation, which is not so high maybe due to lack of corresponding speech. So the next step is to record the PP corpus and then proofread the annotation according to the speech reference. In addition, this work only adopts the objective evaluation criteria. The prediction results inconsistent with the annotation are considered absolutely wrong. However, there may be more than one acceptable phrasing pattern for the same sentence. The prediction results inconsistent with the annotation may be acceptable in the sense of hearing. Therefore, for PP break prediction, the subjective evaluation seems to be more reasonable, but much more laborious than the objective evaluation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editors for their constructive criticism and comments.

REFERENCES

[1] P. Taylor and A. W. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, vol. 12, pp. 99–117, 1998.

[2] C. W. Wightman and M. Ostendorf, "Automatic labeling of prosodic patterns," *IEEE Trans. on Speech and Audio Processing*, vol. 2, pp. 469–481, 1994.

[3] S. Zhao, J. H. Tao and L. H. Cai, "Rule-learning based prosodic structure prediction," *Journal of Chinese Information Processing*, vol. 16, pp. 30–37, 2002.

[4] J. F. Li, G. P. Hu and R. H. Wang, "Prosody phrase break prediction based on maximum entropy model," *Journal of Chinese Information Processing*, vol. 18, pp. 56–63, 2004.

[5] E. Brill, "Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging," *Computational Linguistics*, vol. 21, pp. 543–565, 1995.

[6] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *Natural language processing using very large corpora*, vol. 11, S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann and D. Yarowsky, Eds. Dordrecht: Kluwer Academic Publishers, 1999, pp. 82–94.

[7] E. Brill, "Learning to parse with transformations," in *Text, Speech and Language Technology*, vol. 1, H. Bunt and M. Tomita, Eds. Dordrecht: Kluwer Academic Publishers, 1996, pp. 221–240.

[8] J. K. Low, H. T. Ng and W. Guo, "A maximum entropy approach to Chinese word segmentation," in Proc. SIGHAN Workshop on Chinese Language Processing (SIGHAN 2005), Oct. 2005, pp. 161–164.

[9] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP 96), May 1996, pp. 133–142.

[10] H. T. Dang, C. Chia, M. Palmer and F. Chiou, "Simple features for Chinese word sense disambiguation," in Proc. International Conference on Computational Linguistics (COLING 2002), Aug. 2002, pp. 88–94.

[11] A. L. Berger, S. A. D. Pietra and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, pp. 39–71, 1996.

[12] F. Z. Liu, "Research on Chinese prosodic structure prediction approaches," Ph.D. Dissertation, Institute of Automation, Chinese Academy of Sciences, 2009.

[13] A. Ratnaparkhi, "Maximum entropy models for natural language ambiguity resolution," Ph.D. Dissertation, University of Pennsylvania, 1998.

[14] D. Yarowsky, "Hierarchical decision lists for word sense disambiguation," *Computers and the Humanities*, vol. 34, pp. 179–186, 2000.

[15] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in Proc. International Conference on Machine Learning (ICML 97), Jul. 1997, pp. 412–420.

[16] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and naive Bayes," in Proc. International Conference on Machine Learning (ICML 99), Jun. 1999, pp. 258–267.

Fangzhou Liu received the B.S. degree in information engineering from Zhejiang University in 2003, the M. S. and Ph.D. degrees in 2006 and 2009 respectively, both in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, China.

He is currently a Lecturer in the College of Mathematics and Computer Science at Hunan Normal University, where he has been since 2009. His research interests include text-to-speech synthesis and natural language processing.

You Zhou received the B.S. and M.S. degrees in 2006 and 2009 respectively, both in applied mathematics from Bejing Normal University, Beijing, China.

She is currently a Teaching Assistant in the Department of Applied Mathematics at Hunan University of Finance and Economics, which she joined in 2009. Her research interests include machine learning and pattern recognition.