INTRODUCTION

Prosodic phrasing or prosodic phrase prediction plays an important role in improving the naturalness and intelligence of TTS systems. Linguistic research shows that the utterance produced by human is structured in a hierarchy of prosodic units, including phonological phrase, intonation phrase and utterance [1]. Prosodic structure makes the utterance sound natural and sometimes can help resolving syntactic ambiguity. But the output of syntactic analysis in TTS framework is often a structure of syntactic units, such as words or phrases, which are usually not equivalent to the prosodic ones. Therefore the object of prosodic phrasing is to map syntactic structure into prosodic counterpart.

A lot of methods have been introduced to predict prosodic phrase in English text. These methods are mainly data-driven based procedure such as Classification and Regression Tree (CART) [2], Hidden Markov Model (HMM) [3], neural network autoassociators [4]. For Chinese prosodic phrasing, the traditional method is based on handcrafted rules. And Recurrent Neural Network (RNN) [5] as well as part-of-speech (POS) bi-gram and CART based methods [6] is experimented recently. However, due to the difference in training corpus and evaluation methods between researchers, the results are generally less comparable.

ABSTRACT

Prosodic phrasing is an important component in modern TTS systems, which inserts natural and reasonable break in long sentences. This paper reports the study of applying several inductive machine-learning algorithms to prosodic phrasing in unrestricted Chinese texts. Two feature sets are carefully selected considering the effectiveness and reliability in practice. Then features and the target boundary labels are extracted from a prepared speech corpus, and are used as training examples for inductive learning algorithms including decision tree (C4.5), memory-based learning (MBL) and support vector machines (SVMs). The paper places emphasis on the comparison of the performance and speed of different learning techniques by training and testing them on the same corpus. The experiments show that all the algorithms achieve comparable results for both prosodic word and phrase prediction. It seems that prosodic word can be predicted from Chinese texts more accurately than prosodic phrase when using the same features and learning technique. Inductive learning is a promising way to prosodic phrasing, but it’s more important to find out good feature sets than to apply different learning algorithms in order to improve the prediction accuracy dramatically.

This paper explores prosodic phrasing with three different inductive machine-learning techniques. Features together with the boundary labels are collected at each word boundary from a speech corpus to establish training and testing sets, which are used to experiment with inductive classifiers based on decision tree (C4.5), memory-based learning (MBL) and support vector machines (SVMs). The paper is organized as follows. Section 2 describes the methodology, feature selection, and evaluation methods for prosodic phasing. Section 3 covers the application of three learning algorithms to our problem. Section 4 reports the experiments. The results of different methods are given in section 5.

2. PROSODIC PHRASING

2.1. Prosodic Phrasing Methodology

It has been showed that Chinese utterance is also structured in a prosodic hierarchy, in which there are mainly three levels of prosodic units: prosodic word, prosodic phrase and intonation phrase [7]. Since intonation phrase is usually indicated by punctuation marks, what we have to consider is the prediction of prosodic word and phrase. Figure 1 shows the prosodic structure of a Chinese sentence. In the tree structure, the non-leaf nodes are prosodic units and the leaves are syntactic words. A prosodic phrase is composed of several prosodic words, each of which in turn consists of several syntactic words.

Suppose we have a string of syntactic words i.e. $w_1, w_2, \ldots, w_n$, the boundary between two neighbouring words $w_i, w_{i+1}$ is the object to be studied. There are total three types of boundaries, which can be labelled as $B_0$ (the words are in the same prosodic word), $B_1$ (the words are in the same prosodic phrase, but not the same prosodic word), or $B_2$ (the words are in different prosodic phrases). Assume the label of a boundary is determined by its contextual linguistic information represented by a feature vector $\hat{F}$, prosodic phrasing can be viewed as a classification problem that in
essence can be handled with any trained classifiers, taking the feature vector \( \vec{F} \) as input and giving the most probable boundary label as output.

What we aim to build is a robust prosodic phrasing module that can be embedded in real-time TTS systems. Figure 2 shows the position of prosodic phrasing in the whole TTS framework. When the system is running, syntactic analysis modules provide linguistic feature vectors to prosodic phrasing module. For Chinese text, syntactic analysis may include word segmentation, part-of-speech tagging and sentence parsing. Every analysis stage is not fully accurate and will introduce noise to its following procedure. Thus the prosodic phrasing module should be robust to noisy inputs. To achieve the goal, we not only need robust machine learning algorithms but also should use noisy training data to adapt them to the requirement.

![Figure 2: Prosodic phrasing in TTS framework](image)

### 2.2. Features Selection

Linguistic information around word boundary is the main source of features. The features may come from different levels including syllable, word, phrase, and sentence level. And the type of features can be phonetic, lexical, syntactic, and semantic. Which features have most close relation with prosodic phrasing and how to represent them are still open research problem. A good feature set can help to improve the prediction accuracy but the design of it is usually work intensive and needs much linguistic experience [8].

Another consideration about feature selection is that the selected features could be retrieved much reliably and efficiently in real-time circumstances. Part-of-speech (POS) sequences are the most popular features used in the previous research. And it’s much easier to automatically get POS tags from unrestricted Chinese text than other deep syntactic structures such as syntactic phrase or components. Due to this fact, we design two feature sets based on POS features. One is a base feature set (BFEATS), using a boundary label history of \( B_0 \) to predict the label of each boundary. Then the predicted labels can be compared with labels given by human, which are thought to be true, to get a confusion matrix shown in table 1.

Table 1: Confusion matrix

<table>
<thead>
<tr>
<th>True labels</th>
<th>Predicted labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_0 )</td>
<td>( C_{00} )</td>
</tr>
<tr>
<td>( B_1 )</td>
<td>( C_{10} )</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>( C_{20} )</td>
</tr>
</tbody>
</table>

The other feature set (AFEATS) is based on BFEATS, which includes some additional features: (1) the length of each word in the POS window, in Chinese characters; (2) the length of the sentence, in words and Chinese characters; (3) the position from the current boundary to the start and end of the sentence, in words and Chinese characters; (4) the distances from the current boundary to the first pervious break on non-break boundary, in words and Chinese characters. These features are all numeric and related to length or distance. The AFEATS set has \( 20 + 5 + 2 + 2 \times 2 + 2 \times 2 = 35 \) features.

### 2.3. Evaluation Parameters

Prosodic phrasing can be evaluated with subjective or objective measure. The subjective measure is generally performed by perceptive tests, which are undoubtedly convincing but time-consuming to conduct on large corpus. In this paper, only the objective measure is adopted. As a classification task, prosodic phrase prediction should be evaluated with consideration on all the boundary labels. The trained classifiers are applied on a test corpus to predict the label of each boundary. Then the predicted labels are compared with labels given by human, which are thought to be true, to get an evaluation parameters shown in table 1.

\[
\begin{align*}
\text{Rec}_i &= C_{ii} / \sum_{j=0}^{2} C_{ij} \quad (i = 0,1,2) \\
\text{Pr}_{c_i} &= C_{ii} / \sum_{i=0}^{2} C_{ij} \quad (i = 0,1,2) \\
F_i &= 2 \times \text{Rec}_i \times \text{Pr}_{c_i} / (\text{Rec}_i + \text{Pr}_{c_i}) \quad (i = 0,1,2) \\
\text{Acc} &= \frac{1}{2} \sum_{i=0}^{2} C_{ii} / \sum_{i=0}^{2} \sum_{j=0}^{2} C_{ij} \\
\text{Rec}_c &\text{ defines the recall rate of boundary label } B_c, \text{ while } \text{Pr}_{c_i} \text{ defines the precision rate of } B_c. \text{ Since the counts of different boundary labels are usually unbalanced in the corpus, } F_i \text{ is used as a combination of the recall and precision rate } [9]. \text{ Acc is the overall accuracy of all the labels. If the number of labels is reduced to two, the evaluation parameters can be deduced similarly.}
\end{align*}
\]

### 3. INDUCTIVE LEARNING CLASSIFIERS

A classifier is a function that maps the input feature vector \( \vec{F} = (x_1, x_2, ..., x_n) \) to a confidence that the input belongs to a class. In the case of prosodic phrasing, the features are from linguistic information around the boundary and the classes are the boundary labels. In this paper we use and compare three typical inductive learning methods: decision tree, memory based learning and support vector machines.
3.1. Decision tree learning

A decision tree is grown from the training data using C4.5 decision tree algorithm [10], which is robust to noise and can handle both numeric and symbolic features automatically. When classifying, a class probability rather than a single class label is obtained at the leaf nodes. As a simple strategy, we select the class with the maximum probability as the predicted label.

3.2. Memory based learning

Unlike decision tree learning, the memory-based method is a “lazy” learning scheme originated from k-NN classifier [11]. It stores all the examples presented for training in a structured memory (table or tree) and doesn’t make any further abstraction. During testing, a feature vector of an unseen example is presented. Its distance to all examples in the memory is computed using a similarity metric and the label of the most similar instances is used as the predicted label.

In our problem, there are both symbolic and numeric features. The Overlap metric is selected to compute distance between feature vectors. For feature vectors X and Y, the distance between them can be formulated as

\[ \Delta(X, Y) = \sum \delta(x_i, y_i) \]

where \( \delta(x_i, y_i) \) is 0 if \( x_i = y_i \) and 1 otherwise.

3.3. Support vector machines

Support vector machines are learning techniques based on statistical learning theory. The original idea of SVM is to find a hyperplane that separates the training data into two classes, the margin \( d \) between which is maximized by the hyperplane.

\[ \text{Support vectors} \]

\[ \text{Optimum hyper plane} \]

\[ \text{Figure 3: Support vector machine: the linear separable case} \]

If training data cannot be linearly separated, SVMs use two methods to handle this difficulty. First, it allows classification errors. Second, it non-linearly transforms the training data to a higher dimension feature space, where the data would be more possible to be separated linearly. The hidden feature space is associated with input space by kernel functions. LIBSVM [12], a library of support vector learning implementing efficient algorithms to find the optimum hyper plane, is used as our experiment tool.

In the case of prosodic phrasing, we test linear, polynomial, radial based kernels only to find the linear kernel performs the best. The numeric feature values are normalized into \([0, 1]\) before training. As to the symbolic features, a new coding scheme is applied instead of the common one-of-C coding. For each symbolic feature value \( b \) of the feature \( x \), it is denoted as a \( C \)-dimensional vector of conditional probabilities.

\[ X = (P(B_1 | x = b), P(B_2 | x = b), ..., P(B_C | x = b)) \]

\( C \) is class number and \( B_i \)s are class labels. Such kind of coding creates more compact input vectors of smaller dimension to SVMs than one-of-C coding. The experiments prove that it improves the classification accuracy and shortens the training time.

4. EXPERIMENTS

4.1. The corpus

In our experiments, a speech corpus for our TTS system is used for training and testing. The corpus has 1000 long sentences, which are randomly chosen from newspaper and read by a radiobroadcaster. Two experienced annotators label the sentences with two-level prosodic boundaries by listening to the record speech. The labeling results of them achieve a high consistency rate of 98.5%. There are 19800 Chinese characters in the corpus, which constitute 13375 words. The number of prosodic word boundaries (B1) is 3900 and that of prosodic phrase ones (B2) is 4135.

The sentences of the corpus are also processed with a text analyzer, where Chinese word segmentation and part-of-speech tagging are accomplished in one step using a statistical language model. The segmentation and tagging yields a gross accuracy rate over 94%. The output of the text analyzer is directly used as the training data of learning algorithms without correcting segmentation or tagging errors because we want to train classifiers on noisy data from the real situation.

4.2. Phrasing experiments

There are three boundary classes (B0, B1, B2) in the corpus, the prediction of which is a multi-class classification task. To simplify the problem, we merge the three classes into two since in most systems only one prosodic level is used to generate target pitch contours. It’s possible to merge B0, B1 into a class B10, or merge B1, B2 into a class B12, which gives rise to two different classification problems. The former one is prosodic word prediction; the latter is prosodic phrase prediction. For learning algorithms, the main difference is that the training data of prosodic phrase prediction is more heavily unbalanced than that of prosodic word prediction.

To estimate the generalization ability of a learning algorithm, we apply five-fold cross validation test on the corpus to obtain the generalized results. The corpus data is divided equally into five portions. At each step we train the algorithms on four portions and test them on the rest one. Since the feature sets proposed in section 2.2 include a boundary label history, which are unknown in testing data, the evaluation module of the learning tools is modified to enable the extraction of feature vectors at each boundary in a sentence as the test goes on from left to right.

5. RESULTS

5.1. Training and Testing Time

We train and test the three inductive classifiers on the same data
set that has 12375 training samples. The last word of each sentence is not considered since there is always a break after it. The C4.5 experiment is the fastest one because the divide-and-conquer strategy for learning and the classification with the tree are both quick. The MBL experiment is a bit lower than C4.5 learning. Although the training of MBL is very quick, the computation of the distance from the new example to all the stored ones. The SVM experiment is the slowest, because it is slow owing to the learning. Although the training of MBL is very quick, the SVM experiment is the slowest, because it needs too much floating computation for training and testing.

5.2. Classification accuracy

Table 2 shows the recall rate, precision rate and F-measure of the experiments according to section 2.3. When predicting prosodic word, the use of BFEATS feature set results a low accuracy rate (around 78%) for all the classifiers. But the recall, precision and accuracy rate all improves much if AFEATS features are used instead. When predicting prosodic phrase, the choosing of BFEATS or AFEATS makes little difference in the final results. The performance between classifiers is comparable. The evaluation parameters are close respectively when the features and classes used are identical. The best accuracy rate for prosodic word prediction is 88.4%, got by MBL method, and that for prosodic phrase prediction is 83.4%, got by SVM.

### Table 2: Results of comparative experiments using different methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Features</th>
<th>Classes</th>
<th>Recc</th>
<th>Prec</th>
<th>F</th>
<th>Recf</th>
<th>Precf</th>
<th>Ff</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.946</td>
<td>0.839</td>
<td>0.894</td>
<td>0.429</td>
<td>0.728</td>
<td>0.540</td>
<td>0.815</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.974</td>
<td>0.841</td>
<td>0.989</td>
<td>0.470</td>
<td>0.750</td>
<td>0.578</td>
<td>0.826</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.640</td>
<td>0.826</td>
<td>0.727</td>
<td>0.896</td>
<td>0.767</td>
<td>0.827</td>
<td>0.787</td>
</tr>
<tr>
<td>MBL</td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.920</td>
<td>0.842</td>
<td>0.979</td>
<td>0.491</td>
<td>0.676</td>
<td>0.569</td>
<td>0.811</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.945</td>
<td>0.851</td>
<td>0.989</td>
<td>0.463</td>
<td>0.742</td>
<td>0.570</td>
<td>0.823</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.656</td>
<td>0.816</td>
<td>0.727</td>
<td>0.887</td>
<td>0.713</td>
<td>0.826</td>
<td>0.787</td>
</tr>
<tr>
<td>SVM</td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.913</td>
<td>0.888</td>
<td>0.949</td>
<td>0.922</td>
<td>0.882</td>
<td>0.906</td>
<td>0.894</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.959</td>
<td>0.838</td>
<td>0.960</td>
<td>0.466</td>
<td>0.721</td>
<td>0.566</td>
<td>0.871</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.925</td>
<td>0.862</td>
<td>0.983</td>
<td>0.565</td>
<td>0.719</td>
<td>0.632</td>
<td>0.834</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.633</td>
<td>0.822</td>
<td>0.715</td>
<td>0.896</td>
<td>0.763</td>
<td>0.824</td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td>AFEATS</td>
<td>B0, B2</td>
<td>0.936</td>
<td>0.868</td>
<td>0.949</td>
<td>0.922</td>
<td>0.882</td>
<td>0.906</td>
<td>0.894</td>
</tr>
</tbody>
</table>

6. DISCUSSION

According to the experiments, the prediction of prosodic word achieves better results than that of prosodic phrase. The recall and precision rate of prosodic word boundary can be over 81%, while prosodic phrase has best recall rate of 56.5% by SVM and precision rate of 74.2% by MBL. The difference can be demonstrated that all the classifiers can achieve comparative results but more effective features still need to be studied to improve the prediction accuracy dramatically.

7. CONCLUSIONS

In this paper, we explore the application of three inductive machine-learning methods to the prosodic phrasing problem. Classifiers are trained and tested on the same dataset. The results demonstrate that all the classifiers can achieve comparative results but more effective features still need to be studied to improve the prediction accuracy dramatically.

8. ACKNOWLEDGEMENTS

This work is supported by Hi-Tech Research and Development Program of China (863 program No: 2001AA14072). And thanks to the authors of the tools we used for their great efforts.

9. REFERENCES