PROSODIC PHRASING WITH INDUCTIVE LEARNING

[†]*Zhao Sheng* [‡]*Tao Jianhua* [§]*Cai Lianhong*

Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
[†]`szhao00@mails.tsinghua.edu.cn` {[‡]`jhtao`, [§]`clh-dcs}@tsinghua.edu.cn`

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 comparative 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.

1. 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.

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.

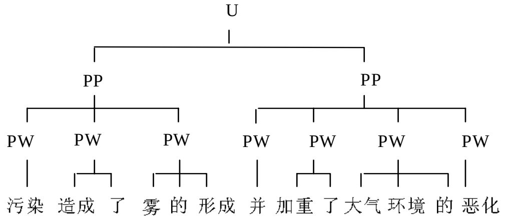

Figure 1: Two-level prosodic structure tree (U for intonation phrase, PP for prosodic phrase, PW for prosodic word)

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 (w_i, w_{i+1} 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 $\vec{F}$, prosodic phrasing can be viewed as a classification problem that in essence can be handled with any trained classifiers, taking the feature vector $\overrightarrow{F}$ 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.

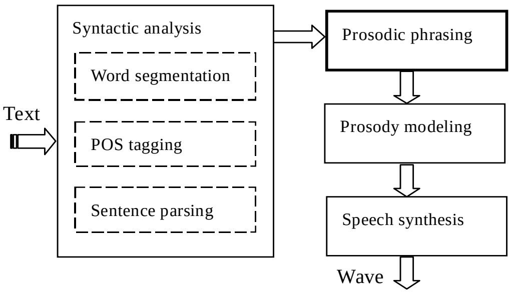

Figure 2: Prosodic phasing in TTS framework

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 previous five words and a POS window of five-word width, three to the left and two to the right of the boundary. The size and position of the POS window is determined according to some elementary tests. We use POS features from three POS sets simultaneously. The first one is the POS set of the tagger having 30 POS tags. The second one is much larger, in which the top 100 frequent words themselves are treated as independent POS tags in addition to those in the first set. The last one has only two tags: content words or functional words. The content words are those belonging to POS tags that are open word set. The functional words are on the contrary. The adoption of multiple POS sets results in POS features of different granularity. Finally the *BFEATS* set has total 5 + 3 * 5 = 20 features, all of which are symbolic values.

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 or 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 * 2 + 2 * 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 a confusion matrix shown in table 1.

True labels	Predicted labels		
	B₀	*B₁*	*B₂*
B_0	C_{00}	C_{01}	C_{02}
B_1	C_{10}	C_{11}	C_{12}
B_2	C_{20}	C_{21}	C_{22}

Table 1: Confusion matrix

C_{ij}s are the counts of boundaries whose true label are B_i but predicted as B_j. From these counts, we can deduce the evaluation parameters for prosodic phrasing.

$$\mathrm{Re}c_i = C_{ii} / \sum_{j=0}^{2} C_{ij} \qquad (i = 0,1,2) \tag{1}$$

$$\mathrm{Pr}e_i = C_{ii} / \sum_{j=0}^{2} C_{ji} \qquad (i = 0,1,2) \tag{2}$$

$$F_i = 2 * \mathrm{Re}\,c_i * \mathrm{Pr}\,e_i / (\mathrm{Re}\,c_i + \mathrm{Pr}\,e_i)(i = 0,1,2) \tag{3}$$

$$Acc = \sum_{i=0}^{2} C_{ii} / \sum_{j=0}^{2} \sum_{i=0}^{2} C_{ij} \tag{4}$$

$\mathrm{Re}c_i$ defines the recall rate of boundary label B_i, while $\mathrm{Pr}\,e_i$ defines the precision rate of B_i. Since the counts of different boundary labels are usually unbalanced in the corpus, F_i is used as a combination of the recall and precision rate [9]. 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.

3. INDUCTIVE LEARNING CLASSIFIERS

A classifier is a function that maps the input feature vector $\overrightarrow{F} = (x_1, x_2, ..., x_n)$ to a confidence that the input belongs to a class. In the case of prosodic phasing, 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 $\Delta(X,Y)=\sum_{i=1}^{n}\delta(x_i,y_i)$. If x_i, y_i are numeric values, $\delta(x_i,y_i)=abs((x_i-y_i)/(\max_i-\min_i))$; If they are symbolic values, $\delta(x_i,y_i)=\begin{cases}0 & x_i=y_i\\ 1 & x_i\neq y_i\end{cases}$

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 hyper plane to separate the training data into two classes, the margin *d* between which is maximized by the hyper plane.

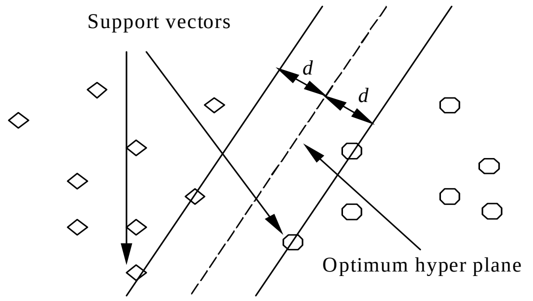

Figure 3: Support vector machine: the linear separable case

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_i, it is denoted as a *C*-dimensional vector of conditional probabilities.

$$X=(P(B_0\mid x_i=b),P(B_1\mid x_i=b),...,P(B_c\mid x_i=b))$$

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 (B_1) is 3900 and that of prosodic phrase ones (B_2) 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 (B_0, B_1, B_2) 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 B_0, B_1 into a class B_{01}, or merge B_1, B_2 into a class B_{12}, 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

Methods	*Features*	*Classes*	$\mathrm{Re}c_0$	$\mathrm{Pr}e_0$	F_0	$\mathrm{Re}c_1$	$\mathrm{Pr}e_1$	F_1	Acc
C4.5	*BFEATS*	B_{01}, B_2	*0.946*	*0.830*	*0.884*	*0.429*	*0.728*	*0.540*	*0.815*
	AFEATS	B_{01}, B_2	*0.947*	*0.841*	*0.890*	*0.470*	*0.750*	*0.578*	*0.826*
	BFEATS	B_0, B_{12}	*0.640*	*0.826*	*0.721*	*0.898*	*0.767*	*0.827*	*0.787*
	AFEATS	B_0, B_{12}	*0.830*	*0.862*	*0.846*	*0.912*	*0.890*	*0.900*	*0.879*
MBL	*BFEATS*	B_{01}, B_2	*0.920*	*0.842*	*0.879*	*0.491*	*0.676*	*0.569*	*0.811*
	AFEATS	B_{01}, B_2	*0.945*	*0.838*	*0.889*	*0.463*	***0.742***	*0.570*	*0.823*
	BFEATS	B_0, B_{12}	*0.656*	*0.816*	*0.727*	*0.887*	*0.773*	*0.826*	*0.787*
	AFEATS	B_0, B_{12}	*0.813*	*0.888*	*0.849*	*0.932*	*0.882*	*0.906*	***0.884***
SVM	*BFEATS*	B_{01}, B_2	*0.939*	*0.838*	*0.886*	*0.466*	*0.721*	*0.566*	*0.819*
	AFEATS	B_{01}, B_2	*0.925*	*0.862*	*0.893*	***0.565***	*0.719*	*0.632*	***0.834***
	BFEATS	B_0, B_{12}	*0.633*	*0.822*	*0.715*	*0.896*	*0.763*	*0.824*	*0.782*
	AFEATS	B_0, B_{12}	*0.816*	*0.847*	*0.832*	*0.902*	*0.881*	*0.891*	*0.868*

Table 2: Results of comparative experiments using different methods

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, classifying a new example with MBL is slow owing to the computation of the distance from the new example to all the stored ones. 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 *AFEAT*S 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.

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 explained as follows. Since prosodic word is the smallest prosodic unit in the prosodic structure, it has more relation with the word level features including word POS, word length etc. Prosodic phrase is larger prosodic unit and cannot be predicted accurately by all the classifiers using *BFEATS* or *AFEATS* features. Word level features are not enough for prosodic phrase prediction, thus syntactic features that dominate one or more words may be used to improve the results [8].

However, it's difficult to compare our results with those reported in [5] [6]. This is because of two reasons. On one hand all the sentences in our corpus have more than 10 Chinese characters and haven't any punctuation inside them. Such kind of sentences is more complex and difficult to handle than regular one. On the other hand our results are based on cross validation tests, which give a better estimation of the performance when the classifiers are running on noisy inputs.

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: 2001AA114072). And thanks to the authors of the tools we used for their great efforts.

9. REFERENCES

[1] Abney Steven. Chunks and dependencies: bringing processing evidence to bear on syntax. Computational Linguistics and Foundations of Linguistic Theory, CSLI, 1995

[2] Michelle Wang and Julia Hirschberg. Automatic classication of intonational phrase boundaries. Computer Speech and Language 6:175–196, 1992.

[3] Paul Taylor, Black and Alan.W.Black. Assigning phrase breaks from part-of-speech sequences, Computer Speech and Language v12, 1998.

[4] Achim F. Muller, Hans Georg Zimmermann and Ralph Neuneier, Robust generation of symbolic prosody by a neural classifier based on autoassociators, ICASSP2001

[5] Zhiwei Ying and Xiaohua Shi. An RNN-based algorithm to detect prosodic phrase for Chinese TTS, ICASSP2001

[6] Yao Qian, Min Chu, Hu Peng. Segmenting unrestricted chinese text into prosodic words instead of lexical words, ICASSP2001

[7] Li Aijun, Lin Maocan. Speech corpus of Chinese discourse and the phonetic research. ICSLP200

[8] Julia Hirschberg, Owen Rambow. Learning Prosodic Features using a Tree Representation. EuroSpeech 2001

[9] C.J. van Rijsbergen. Information Retrieval. Butterworths, London. 1979.

[10] Quinlan,J.R. Induction of decision trees. Machine Learning, 1(1):81-106, 1986

[11] Walter Daelemans, Jakub Zavrel. TiMBL: Tilburg Memory Based Learner, version 4.0, Reference Guide. ILK Technical Report 01-04, 2001.

[12] Chih-Chung Chang and Chih-Jen Lin.LIBSVM : a library for support vector machines. 2001.