Combining Length Distribution Model with Decision Tree in Prosodic Phrase Prediction

Qin Shi, Jiang DanNing, Meng FanPing,Qin Yong
IBM China Research Lab, Beijing, China

E-mail: { shiqin,jiangdn,mengfp,qinyong } @cn.ibm.com

ABSTRACT

In Text-to-Speech (TTS) systems, prosody phrase prediction is important for the naturalness and intelligibility of synthesized voice. Statistic methods, such as dynamic programming (DP), decision tree (DT), maximum entropy (ME), etc, have been considered for the task. Features based on syntactic and lexical information are widely used. However, the predicted prosody phrases are often observed to have unrealistic length due to the lack of length distribution modeling. This paper proposes a novel algorithm to incorporate the length distribution model in prosody phrase prediction. Rather than directly use phrase length as a feature of DT or ME, the algorithm exploits the correlation between the length and the possibility given by a decision tree. Experiments show that the recalling rate and precise rate are improved 16.37% and 14.05% relatively by using the proposed algorithm.

10.21437/Interspeech.2007-217

1. INTRODUCTION

Prosody generation can be divided into two levels: symbolic prosody and phonetic prosody. The symbolic prosody is often carried out as a part of text analysis by the linguistic modeling component, which is also referred as the TTS front-end. Prosody symbolic information includes break index, accent, and etc. It substantially influences the phonetic prosody parameter in a TTS system. As an important component of the prosody symbolic information, break index of prosody phrase largely affects the naturalness and intelligibility of synthetic voices. Therefore, break index prediction is vital for a TTS system. Although there is a tight relationship between syntax and break index, the syntactic and lexical information does not completely determine the prosodic structure of a sentence. Prosody structure is also influenced by speaker's pronunciation habits, which can be extracted from speech corpus.

Statistical approaches for prosodic structure prediction have been widely used for capturing the complex relationship between prosodic and syntactic structures, such as dynamic programming, decision tree, and maximum entropy [1]-[3]. Usually, both syntactic information and lexical information are included as features. However, the prosody phrase length information, which contains important information about the speaker's pronunciation habits, is not effectively used in these methods.

Motivated by the above considerations, we propose a new algorithm to combine prosody phrase length distribution with the possible candidates given by the decision tree to determine the prosody phrase boundary. For training, a manually labeled corpus is used.

The remainder of the paper is structured as follows: Section 2 introduces the new algorithm that incorporates prosody phrase length distribution to determine the prosody phrase boundary. Section 3 discusses the experiments in detail, which is followed by the conclusion.

2. PROSODY STRUCTURE PREDICTION

In this section， we first describe the Mandarin linguistic analysis components and the labeled corpus. Then, the decision tree and the related features for prosody structure prediction are discussed. Subsequently, the proposed algorithm is introduced.

2.1 Introduction of Mandarin TTS Front-End

There are mainly two tasks in current TTS Front-End: one is grapheme-to-phoneme conversion, the other is prosody symbolic prediction. Both tasks are based on basic linguistic analysis. In mandarin TTS systems, this basic linguistic analysis includes: lexical word segmentation, part of speech (POS) tagging, semantic analysis, and syntactic structure analysis.

The techniques for lexical word segmentation and POS tagging are more mature than semantic tagging and syntactic structure analysis. Figure 1 shows the components of linguistic analysis in a typical concatenative Mandarin TTS system.

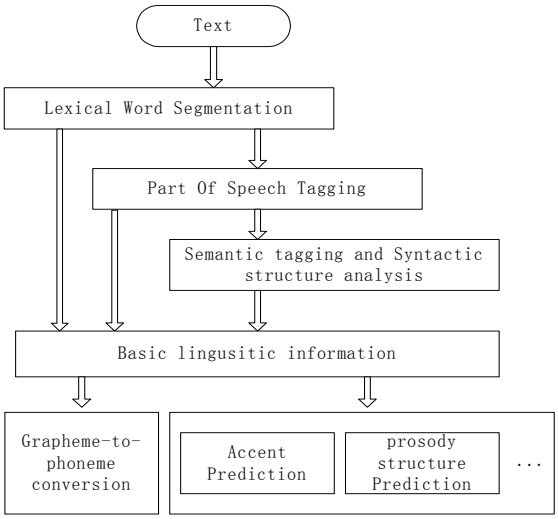

Fig. 1 Linguistic Analysis in Mandarin TTS System

2.2 Manually labeled Mandarin Speech Corpus

For a concatenative TTS system, the quality of speech corpus is very important, it is highly dependent on scripts design, speaker selection, recording quality, alignment, and prosody labeling quality.

A description of prosodic events is obtained by a manual annotation of prosodic structures after recording. The principle of annotation is "to label what you hear" [4]. For Mandarin Chinese, the prosody structure is usually defined by the following hierarchy: foot, prosody word, prosody phrase and intonation phrase.

The perceptive cues used as labeling judgment include pitch discontinuity, change of rhythm, length of pause, and duration lengthening. Punctuation can also be used for the prediction of intonation phrases. However, our work so far has been focused on the prediction of prosodic word and prosody phrase.

2.3 Prosody phrase boundary prediction using decision tree

Prosody phrase structure is more flexible than prosody word and intonation phrase. As a well-known machine learning method, decision tree is a way to represent rules underlying data with hierarchical and sequential structure that recursively partition the data. Decision trees automatically constructed from data have been used successfully in many real-world situations. Their effectiveness has been compared widely to other automated data exploration methods and to human experts. The algorithm is an appropriate choice for prosody phrase prediction as well.

In the prosody phrase prediction task, the problem can be described as follows: Given a sequence *U* of prosody word units: $U = \{u_1,..., u_n\}$, every unit will have a corresponding feature vector x_i, and a parameter $a_i \in \{0,1\}$, indicating whether the unit is prosodic phrase boundary or not.

During the construction of the binary decision tree, each tree node is assigned with one question, and the candidate questions are pre-designed that are associated with the features capturing context information in the text. In run time, each unit will have a feature vector, and the vector traverses the decision tree (D_T) according to the questions until a leaf is reached. The parameter distribution of the leaf determines whether the unit's right boundary is the prosody phrase boundary. The possibility of prosody phrase boundary for unit u_i can be defined as:

$$p(a_i \mid u_i) = p(a_i \mid x_i, D_T)$$

Usually a threshold is set to help the final decision of whether there is a prosody phrase boundary after the unit. With the above method, experiments show that if a fixed threshold is used, the hypothesized prosody phrases often will have lengths that are either too long or too short. This in turn destroys the rhythm of the synthesized voice. Thus, it becomes necessary to use a dynamic threshold that takes the prosody phrase length distribution into consideration.

2.4 Combining prosody phrase's length distribution model with Decision Tree

The fundamental idea of the algorithm is to determine the phrase boundaries sequentially in the order of their corresponding probabilities. At the same time, for a boundary to be set, its possibility must be (1) larger than a minimum threshold; and (2) the length to the nearby left and right boundaries that already set should be larger than minimum length limitation. The boundaries generated by the above should be scanned again. If the length between two boundaries is larger than maximum length limitation, a new boundary will be inserted between them. The boundary should have the largest probability in all the candidate boundaries and at the same time satisfies the length constraints. The scanning terminates when all resulting prosody phrases fits the constraints.

The algorithm can be described as follows:

In one sentence, prosody words can be described as: $U = \{u_1,..., u_n\}$. For each prosody word $u_i (1 \le i \le n)$, The possibility of its right boundary to be a prosody phrase boundary is defined as $p_{PP}(u_i)$, which comes from the Decision Tree. The head of the sentence is annotation by u_0 .

The length between two units u_i and u_j can be defined as:

$Len(u_i)$ means the syllable length of prosody word u_i, and

$$Length(i, j) = \sum_{i<k\le j} Len(u_k)$$

Step1:

Set a flag for every prosody word:

$f(u_0) = 1, f(u_n) = 1$

$f(u_i) = 0 \quad (1 \le i < n)$,

In Step1, $f(u_0) = 1$ means the head of a sentence is a boundary. $f(u_n) = 1$ means the end of a sentence is also a boundary.

If $f(u_i) = 0$, u_i 's right boundary is not selected as a prosody phrase boundary yet.

If $f(u_i) = 1$, u_i 's right boundary is already selected a prosody phrase boundary.

Step2:

Do{

$$j = \arg\max_{1 \le i < n \,\&\&\, f(u_i) == 0} (p_{pp}(u_i))$$

if ($p_{pp}(j)$ <0.5) { break;}

Let $m(0 \le m < j)$ be j's nearest left prosody phrase boundary that was already set,

Let $k(j < k \le n)$ be j's nearest right prosody boundary that was already set.

if ($length(m, j)$ > min_threshold && $length(j, k)$ > min_threshold && $length(m, k)$ >max_threshold) {

$f(u_j) = 1$ }

else {

$f(u_j) = -1\}$

} while ($p_{pp}(j) > 0.5$)

In Step2, if one boundary's possibility is larger than 0.5 and the current phrase length is larger than minimal length, the boundary is set. So the overall order by which the boundaries are determined is in accordance with the possibilities' decreasing order.

Step3:

```
for (i=1; i<n; i++){
  if( f(u_i) ==-1){ f(u_i) = 0 }
}
Do {
  Flag_SetBoundary = 0;
  for (i=1; i<n ;i++){
    if( f(u_i) ==1) {
      Let k be i's nearest right boundary
      if( Length(i,k) > max_threshold) {
        l = arg max (p_pp(u_m))
            i≤m<k&&
            length(i,l)≥min_threshold &&
            length(l,k)≥min_threshold
        f(u_l) = 1;
        Flag_SetBoundary = 1;
      }
    }
  }
} while ( Flag_SetBoundary == 1 )
```

In step2, the boundaries with possibilities larger than 0.5 and match the prosody phrase length's criteria are already set. But the length of some prosody phrases is still larger than the acceptable length. They are adjusted in step3.

In Step2 and Step3, two kinds of threshold are used: one is max_*threshold*, the other is min_*threshold*. Whether the thresholds are stationary or related with the boundaries' possibility should be studied. The detailed information is introduced in the following section.

3. EXPERIMENTS

3.1 Training and Testing Corpus Description

A manually labeled corpus is used in the experiment. 15,000 sentences are used as training data and 5,000 sentences are used as testing data. Prosody word, prosody phrase and intonation phrase are labeled in the corpus. [4]

For an example, for the following sentence:

两天展销会成交额达一点一亿元，比上届展销会成交额增加近一倍。

The labels are annotated as: the space character stands for the prosody word boundary, (BP1) stands for the prosody phrase boundary and (BP2) stands for intonation phrase boundary.

The prosody structure will be labeled like the following:

两天 展销会 成交额 (BP1) 达 一点一亿元 (BP2) 比 上届 展销会 (BP1) 成交额 增加 近一倍 (BP2) (1)

And the result of automatic segmentation and POS tagging is:

两(mx) 天(qt) 展销会(ng) 成交额(ng) 达(vgn) 一点一亿(mx) 元(qnm) ，(w2) 比(pg) 上届(ng) 展销会(ng) 成交额(ng) 增加(vgn) 近(ag) 一(mx) 倍(mab)。(w1) (2)

Aligning the above data with the break index, the training data for prosody word prediction will be:

两(mx) 天(qt) ‖ 展销会(ng) ‖ 成交(vg) 额(ng) ‖ 达(vgn) ‖ 一点一亿(mx) 元(qnm) ‖ ，(w2) ‖ 比(pg) ‖ 上届(ng) ‖ 展销会(ng) ‖ 成交(vg) 额(ng) ‖ 增加(vgn) ‖ 近(ag) 一(mx) 倍(mab) ‖ 。(w1) ‖ (3)

Using POS-tag and the number of syllables in a lexical word as features, prosody word can be predicted by combining DP and DT[3].

Based on prosody word layer, the training data for prosody phrase will be:

两天(mx_qni) 展销会(ng) 成交额(vg_ng) ‖ 达(vg) 一点一亿元(mx_qni) ‖ ，(w2) ‖ 比(pg) 上届(ng) 展销会(ng) ‖ 成交额(vg_ng) 增加(vg) 近一倍(ag_mx) ‖ 。(w1) ‖ (4)

Using the POS combination and the syllable number of prosody word as features, decision tree can be built again. For each prosody word, a possibility can be got by traversing DT.

两天(0.040000) 展销会(0.480000) 成交额(0.850000) 达(0.070000) 一点一亿元(1.000000) 比(0.000000) 上届(0.290000) 展销会(0.510000) 成交额(0.850000) 增加(0.220000) 近一倍(1.000000) (5)

In the following section, the correlation between the possibility and prosody phrase length will be studied.

3.2 Distribution of Prosody Phrase Length

If a prosody word's right boundary is a prosody phrase boundary, the boundary's possibility (*poss*) given by decision tree and the length to its nearest left prosody phrase boundary (*leng*) will be recorded. It can be presented as: $v_i(poss, leng)$.

For example: From (4) and (5), we can get four vectors: (0.85, 8), (1.00, 6), (0.51, 6), and (1.00, 8). Among all the candidates' boundaries, the possibility of a candidate boundary to be a real prosody phrase boundary is counted according to different *poss* range and shown in the following table.

poss range	Possibility
0.0~0.1	0.028
0.1~0.2	0.127
0.2~0.3	0.204
0.3~0.4	0.263
0.4~0.5	0.338
0.5~0.6	0.4088
0.6~0.7	0.462
0.7~0.8	0.572
0.8~0.9	0.740
0.9~1.0	0.851
1.0	0.9989

Table 1: Relationship between possibility of DT and Possibility of becoming real prosody phrase boundaries

From the above table, when *poss* is 1.0, it can always be defined as a prosody phrase's boundary, and when a prosody

word's $poss \in [0.0.2)$, because the low possibility, it usually is not a prosody phrase boundary. For example, in all prosody word boundaries whose possibility of DT is in the range of $[0.0,0.1)$, only 0.28% prosody word's boundaries are real prosody phrase boundaries.

With the vector $v_i(possibility, length)$, $c(p,l)$ can be defined as:

$Coun(v_i(poss,leng) | \text{trunc}(poss*10)/10.0==p \&\& leng==l)$

and $0 \le p \le 1.0$, $1 \le l \le 11$

for example : c(5,5) = 932 , means there are 932 vectors whose length is equal 5 and whose possibility is in the range of $[0.5,0.6)$. From the training corpus, prosody phrase length is in the range of $[2,10]$. $c(p,l)$ ($0.2 \le p \le 0.9$) was studied, the following table shows the result:

p\l	2	3	4	5	6	7	8	9	10
0.2	374	806	589	523	371	257	151	59	30
0.3	142	767	767	657	532	322	162	61	36
0.4	61	974	1116	1089	695	501	216	83	40
0.5	98	718	1218	932	650	529	251	117	49
0.6	29	597	951	581	599	419	246	103	45
0.7	120	341	950	1209	1030	802	460	194	83
0.8	25	348	1522	1321	1174	846	625	254	134
0.9	14	129	839	783	878	870	573	287	117

Table 2: Relationship between Prosody Phrase Lengths And Possibilities of Decision Tree

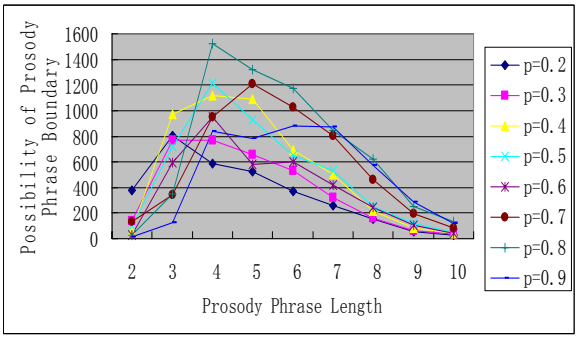

Chart 1: Relationship between Prosody Phrase Lengths And Possibilities of Decision Tree

From the chart, max_length=10. And given prosody phrase possibility (p), the criterion for the length (l) is: $c(p,l)$ should not be low. Given p_i ($0.2 \le i \le 0.9$), min_*threshold* and max_*threshold* can be defined as:

$c(p_i, \text{min_}threshold) \ge Max(c(p_i,l)) * 10\%$

$c(p_i, \text{max_}threshold) \ge Max(c(p_i,l)) * 10\%$

Prosody phrase length range table was showed in the following:

.

p	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Min	2	2	3	3	3	3	3	3
Max	9	9	9	9	9	9	9	10

Table 3: The relationship between possibilities of DT and prosody phrase length range

3.3 Experiment Results

Experiments are carried out to compare the following three methods: Decision Tree without the feature of prosody phrase length, Decision Tree with the feature of prosody phrase length [2] and Combine Phrase length Distribution with Decision Tree. Ignoring the boundaries on the right of the punctuation, the following table shows the result:

Method	Recalling Rate	Precise Rate
DT without PP Length	71.9%	87.9%
DT with PP Length feature	71.8%	88.1%
Combine DT with PP length	76.5%	89.6%

Table3. Experiment Results

Experimental results indicate that prosody phrase information is used more efficiently in the new algorithm than only being treated as a feature in Decision Tree. Compared with DT without the feature of PP length , the recall and precision rates see relative improvements of 16.37% and 14.05%, respectively.

6. REFERENCES

[1] Jian-Feng Li, Guo-Ping Hu, Renhua Wang: "Chinese Prosody Phrase Break Prediction Based on Maximum Entropy Model.", In: ICSLP 2004, Korea

[2] Qin Shi and Volker Fischer : "A Comparison of Statistical Methods and Features for the Prediction of Prosodic Structures", In : ICSLP 2004, Korea

[3] Qin Shi and XiJun Ma: "Statistic Prosody Structure Prediction.", In: *Proc. of the IEEE 2002 Workshop on Speech Synthesis*, Santa Monica, Ca., 2002.

[4] WeiBin Zhu: "Corpus labeling for data driven TTS System.", In: *Proc. of the IEEE 2002 Workshop on Speech Synthesis*, Santa Monica, Ca., 2002

.

[5] C.W. Wightman and M. Ostendorf: "Automatic Labeling of Prosodic Patterns." In: *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Adelaide, 1994.

[6] XiJun Ma: "Automatic Prosody Labeling using both Text and Acoustic Information." In: *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003.