Prosodic Word Boundaries Prediction for Mandarin Text-to-Speech

YanQiu Shao, JiQing Han, Ting Liu, YongZhen Zhao

School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
{yqshao;jqhan}@hit.edu.cn, {tliu;simply}@ir.hit.edu.cn

Abstract

In Mandarin speech, the Prosodic Word (PW) is the basic rhythmic unit instead of Lexical Word (LW), and the naturalness of TTS will be directly influenced by the segmentation of PW. Most of the PWs are the combination of some LWs. In this paper, three models, i.e. a directed acyclic graph (DAG) model, segmentation model and Markov Model (MM) combined with Transformation-Based Error Driven (TBED) learning algorithm are designed to combine lexical words into prosodic words. Considering some long LWs should be broken into two or more PWs, a long word break model is also applied to those LWs. Experimental results show that MM combined with TBED plus a long word break model is the best one among the three methods, and 93.00% precision and 93.23% recall are achieved.

1. Introduction

Prosodic features will directly affect the naturalness of the synthesized speech. Currently, many researchers pay more attention to generating prosody automatically. One of the main obstacles to automatic generation of prosody is identifying the hierarchical prosodic constituents from texts automatically [1].

There is no unified standard of identifying the prosodic hierarchy. Generally, the prosodic hierarchy includes three tiers, which are prosodic word, intermediate phrase and intonational phrase. Prosodic word, which is primarily composed of disyllable or trisyllable, is the most elementary rhythmic unit among these three tiers. In real speech, prosodic word should be uttered continuously and closely without breaks. Intermediate phrase and intonational phrase are the combination of several PWs. Thus the segmentation of prosodic word will affect the segmentation of intermediate phrase and intonational phrase, and will also play an important role in increasing the naturalness of synthesized speech.

In recent years, many methods of predicting prosodic word boundaries have been proposed, such as rule-driven approach, statistical method[2], classification and regression tree (CART) method[3], recurrent neural network (RNN) method[4] and so on. Although the prosodic words do not conform to the lexical words, many studies reveal that there are relationships between PW and LW[1][3]. Now most of the known PW segmentation methods are based on part of speech (POS) features. In this paper, three methods, i.e. DAG model, segmentation model and MM combined with TBED are designed to combine lexical words into prosodic words. Considering a long LW needs to be broken, a break model of long words is also designed. Experimental results show that MM combined with TBED plus the break model of long word gets the best results, and 93.00% precision and 93.23% recall are obtained.

2. Data Preparation

There are 12000 Chinese sentences in the experiment corpus. In the corpus, all the PWs boundaries are labeled manually and all the LWs are segmented and tagged with POS automatically. For example:

(1) 中国球迷盼望着扬眉吐气的那一天。
(Chinese football fans are looking forward to a day when they can hold their head high.)
(2) 中国| 球迷| 盼望着| 扬眉| 吐气的| 那一天。
(3) 中国/nd 球迷/nc 盼望/vg 着/ut 扬眉吐气/i 的/usde 那/r 一/m 天/q 。/wj
(4) 中国/nd | 球迷/nc | 盼望/vg 着/ut | 扬眉/?i | 吐气/?i 的/usde | 那/r 一/m 天/q 。/wj

(1) is an original Chinese sentence, after manual PW annotation, it turns into (2), and after automatic word segmentation and POS tagging it turns into (3). (4) is the combination of (2) and (3), and it is the final form of a sentence in our corpus. The characters behind '/' are the representation of the POS of a lexical word and '/?' represents a part of a Chinese word which is segmented inside. 9000 sentences are selected as training data and the other 3000 are selected as testing data.

3. Statistical models of prosodic word boundaries prediction

From the example in section 2, it can be seen that some PWs are the combination of several LWs. For instance, "盼望着" consists of two LWs, "盼望" and "着". However, some PWs are only a part of a long lexical word, e.g. the lexical word "扬眉吐气" is split into two short PWs, "扬眉" and "吐气". Totally, 132835 LWs and 70032 PWs are obtained in the whole corpus. 5231 PWs are those ones which are parts of LWs words and they cover 7.5% of the whole PWs. The other 92.5% PWs are obtained by combining lexical words. It is obvious that most of the PWs are from the combination of LWs. Therefore, three different models are designed for prediction of PW boundaries as follows.

3.1. DAG model

In order to convert an arbitrary string of LWs $w_1 \ldots w_i \ldots w_L$ into PWs, a directed acyclic graph model is designed which mainly

considers of the probability of the combination of several LWs. Thus some statistical rules are obtained. For example:
(1) r+usde 0.8636 //Pronoun+ Auxiliary
(2) a+ng+used 0.2857 //Adjective+Generalnoun+Auxiliary
……
Item (1) means that the probability of combining a pronoun and an auxiliary into a PW is 86.36%. Item (2) means the probability of combining an adjective, a general noun and an auxiliary into a PW is 28.57%. Totally, 2635 combinational rules are obtained.

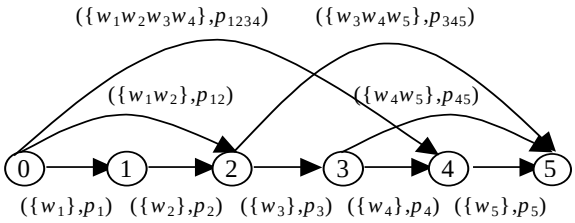

Figure 1: *DAG representation of converting lexical words into prosodic words.*

In fact, the problem of converting a string of LWs $w_1 \ldots w_i \ldots w_L$ which has L lexical words into PWs can be represented by a DAG which has L+1 nodes as shown in Figure 1 (suppose L=5). We can see that each arc has two fields. The first one is POS sequence and the second one is the probability corresponding to the first field from combinational rules. Generally, only if there exists a corresponding probability of POS sequence $w_i w_{i+1} \ldots w_{i+k}$ ($1 \le i \le L$, $0 \le k \le L$-1), the arc from node i-1 to node $i+k$ should be drawn. It should be stressed that if there is no probability of the lexical word between node i-1 and node i, a minimum probability should be given to that arc so that the network could be connective. Thus, the problem of predicting PW boundaries becomes the problem of solving an optimum path from node 0 to node L, and that can be solved by using the dynamic programming algorithm.

3.2. Segmentation model

The DAG model mainly considers the combinational probability of all kinds of POS sequences. However, the most important consideration of segmentation model is whether a segmental tag should be annotated between two lexical words. Suppose that S is the representation of the string of LWs $w_1 \ldots w_i \ldots w_L$, and the corresponding POS sequence of S is $p_1 \ldots p_i \ldots p_L$. Assume that t_i ($t_i \in \{0,1\}$, $i \in \{1,2,\ldots L\text{-}1\}$) shows the segmentation type. If there is a segmentation between p_i and p_{i+1}, then t_i is1, otherwise t_i is 0. The corresponding segmentation type sequence of S is $t_1 \ldots t_i \ldots t_{L\text{-}1}$.

In fact, the function of this model is to decide the value of t_i between p_i and p_{i+1}, and here the position is called segmentation point. The value of t_i has a relationship with the POS sequence before and after the *i-th* segmentation point. Suppose the number of POSs before the *i-th* segmentation point is M and the number of POSs after the *i-th* segmentation point is N. Whether t_i is 0 or 1 depends on the conditional probability $P(t_i | p_{i-M+1} \ldots p_i \ldots p_{i+N})$ which is defined by:

$$P(t_i \mid p_{i-M+1} \cdots p_i \cdots p_{i+N}) = \frac{f(t_i, p_{i-M+1} \cdots p_i \cdots p_{i+N})}{f(p_{i-M+1} \cdots p_i \cdots p_{i+N})} \quad (1)$$

where $P(t_i | p_{i-M+1} \ldots p_i \ldots p_{i+N})$ is the probability of t_i between p_i and p_{i+1}, and $f(t_i, p_{i-M+1} \ldots p_i\, p_{i+1} \ldots p_{i+N})$ is the occurrence times of sequence $p_{i-M+1} \ldots p_i\, t_i\, p_{i+1} \ldots p_{i+N}$, and $f(p_{i-M+1} \ldots p_i\, p_{i+1} \ldots p_{i+N})$ is the occurrence times of POS sequence $p_{i-M+1} \ldots p_i\, p_{i+1} \ldots p_{i+N}$ in the corpus. In our experiment, only p_i and p_{i+1} are considered, that means M equals 1 and N equals 1. Equation (1) is simplified as:

$$P(t_i \mid p_i p_{i+1}) = \frac{f(t_i, p_i p_{i+1})}{f(p_i p_{i+1})} \quad (2)$$

There are 1368 kinds of POS pairs in the training data, and the segmentation model also gives the segmental probability of each kind of POS pair. Here the threshold is 0.5, which means if the segmental probability is over 0.5 then t_i equals 1, and the tag of segmentation will be annotated. Otherwise, t_i equals 0, and the tag will not be annotated.

3.3. MM Model combined with transformation based error driven learning algorithm

3.3.1. MM Model

Some Chinese lexical words occur in fastened positions of prosodic words. For example, "的" usually appears at the beginning of PWs, and prepositions often appear at the end of PWs. Thus the sentences in training corpus are dealt with as follows (taking the example in section 2 for instance):
中国/nd/S | 球迷/nc/S | 盼望/vg/B 着/ut/E | 扬眉/?i/S | 吐气/?i/B 的/used/E | 那/r/B 一/m/I 天/q/I 。/wj/E
Here 'B'(Beginning) represents the beginning of a PW, 'E'(End) is the end of a PW, 'I'(Inside) represents the middle of a PW, and 'S'(Single) means that the PW includes one LW or the PW is only a part of a LW. Then the problem of PW segmentation can be resolved by Markov Model where the observation sequence is a POS sequence $p_1 \ldots p_i \ldots p_L$, and the state sequence is a tag sequence $s_1 \ldots s_i \ldots s_L$ ($s_i \in$ {B, I, S, E}). In order to obtain the state sequence with maximum probability, equation (3) is employed:

$$\begin{aligned} &\arg\max_{s_1 \cdots s_L} P(s_1 \ldots s_L \mid p_1 \ldots p_L) \\ &= \arg\max_{s_1 \cdots s_L} \frac{P(s_1 \ldots s_L) P(p_1 \ldots p_L \mid s_1 \ldots s_L)}{P(p_1 \ldots p_L)} \end{aligned} \quad (3)$$

Since the MM we adopt is first order MM model, equation (3) can be simplified as:

$$\begin{aligned} &\arg\max_{s_1, \ldots s_L} P(s_1 \ldots s_L \mid p_1 \ldots p_L) \\ &= \arg\max_{s_1, \ldots s_L} \prod_{i=1}^{i=L} P(s_i \mid s_{i-1}) P(p_i \mid s_i) \end{aligned} \quad (4)$$

where $P(s_i|s_{i-1})$ is the transition probability and $P(p_i|s_i)$ is the emission probability. All the parameters may be obtained from training data through statistical method, and Viterbi algorithm is used to get the best state sequence.

3.3.2. Transformation based error driven learning algorithm

After the above operations, there are still some segmentation errors. Here TBED learning algorithm is used to correct the remaining errors. This learning algorithm was presented by Brill[5], and has been applied to many areas in NLP. In Figure 2, the process of this algorithm to PW segmentation is illustrated. Firstly, the original text should be initialized, and the results will be regarded as the initial tagged text. Then the tagged text will be compared with standard text in the learning module. If wrong annotation occurs, some candidate rules will be produced according to some rule templates. After this, theses candidate rules will be appended to the ordered rule queue. Whether the rules in the queue will be used as effective rules to the tagged text depends on the estimation function.

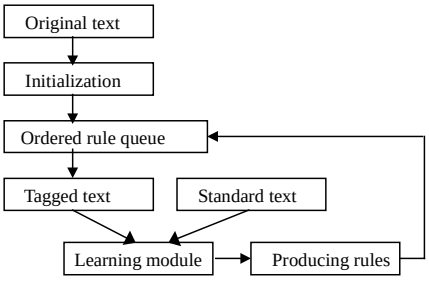

Figure2: *Learning process of transformation based error driven learning algorithm.*

There are three aspects that still need to be described in detail.

(1) Initialization.

Here the PW segmentation results of employing MM model are applied to initialize the original text.

(2) The definition of rule templates.

In experiment, 9 classes of patterns are defined according to the context of the word to be tagged, and the context includes the POSs and word lengths of the two former words and one later word of the current tagged word. The types of patterns are shown below:

If 0: POS = X&1: POS = Y&0: LENGTH = Z -> CHANGE TAG1 To TAG2

This rule template means that if the former POS is *X*, the latter POS is *Y* and the length of the former word is *Z*, change the tag of current word from *TAG1* to *TAG2*. (*TAG1*, *TAG2*∈ {B, I, S, E})

(3) Estimation function.

Suppose a rule 'r' is applied to training corpus. C(r) represents the number of the words which are tagged wrong before using rule 'r' and are corrected after using rule 'r'. E(r) is the number of the words which are tagged right before and are tagged wrong now. The evaluation function F(r) is defined as:

$$F(r)=C(r)-E(r) \quad (5)$$

If $F(r)$ is larger than a given threshold, rule '*r*' will be selected as an effective rule. In this paper, the threshold is 0, and more than 1000 effective rules are obtained.

4. The break model of long words

All the models mentioned above are used to combine LWs into PWs, while some long LWs should be broken into two or more PWs in practice. In this section, the break model is suitable to those long words which need to be broken. Statistical results from the training corpus show that only those LWs which include four or more Chinese characters may be broken into two or more PWs (See Table 1).

Table 1: *The probability of lexical words with different length broken into prosodic words.*

LW length (characters)	Probability of LW broken into prosodic words
2	0.0228
3	0.0806
4	0.8457
5 and more	1.0

So only the probabilities of all kinds of segmentation patterns of these long LWs are recorded. Suppose Break Rule (BR) is an expression as follow:

BR = <POS, WLen, SegP, Prob>

Where *POS* represents the part of speech of a LW. *WLen* is the length of LW. *SegP* represents a kind of segmentation pattern, and *Prob* is the probability of this *SegP*. The elementary segmentation model is the set of all the *break rules*. For example:

BR1=<c, 4, 2+2, 0.8571>

BR1 means the word whose *POS* is *c* and whose length is *4* may be broken into two *2*-character PWs with probability of *0.8571*.

Here another example:

BR2=<c, 4, 3+1, 0.1429>

BR2 means the word whose *POS* is *c* and whose length is *4* may be broken into one *3*-character PW and one *1*-character PW with probability of *0.1429*.

If a LW can be broken into PWs in different segmentation patterns, the maximum probabilistic pattern is always applied. For the example above, since the probability of pattern *2+2* is larger than that of *3+1* and others, the word whose *POS* is *c* and length is *4* will be broken into *2+2* pattern. The rule with POS and length is called special rule. On the other hand, some combinations of POSs and word lengths cannot be found in the training corpus, so the patterns are constructed only according to the word length. For example, all of the 8-character words will be broken into four 2-character PWs. This kind of rules

is called general rule. After selecting special rules from the elementary segmentation model and adding some general rules, 21 patterns of the long word break are gotten in the end.

5. Experiments and discussions

In order to evaluate all the models in this paper, we conducted a series of experiments. Before reporting the experimental results, we first define two evaluation criterions:

Recall (%) = The number of PWs segmented correctly / The number of PWs in the testing set × 100%

Precision (%) = The number of PWs segmented correctly / The number of PWs segmented by machine× 100%

The results of DAG model, Segmentation model and MM combined with TBED are listed in Table 2. It can be seen that the MM combined with TBED achieves the best result. The DAG and Segmentation model approximately have the same precision, however, the DAG model has the lowest recall.

Table 2: *The results of PW segmentation by three algorithms.*

Results / Algorithms	Precision (%)	Recall (%)
DAG Model	86.95	85.54
Segmentation Model	86.82	89.15
MM +TBED	92.89	90.98

DAG model does not achieve the positive result as expected. After analysis, we found that the path with less prosodic words often gets higher probability, thus there is a trend to combine lexical words into prosodic words as much as possible. This is the reason why DAG model obtains lower recall.

After combining with the long words break model, all the three methods obtain higher recall. Averagely, the recall raises about 2.27%. The results are shown in Table 3.

Table 3: *The results of PW segmentation combined with break model of long words.*

Results / Algorithms	Precision (%)	Recall (%)
DAG Model	87.09	87.81
Segmentation Model	86.96	91.42
MM + TBED	93.00	93.23

6. Conclusions

In this paper, we design several models for automatically predicting prosodic word boundaries, and also evaluate these methods by experiments. According to the experimental results we conclude that MM combined with TBED plus long word break model can achieve higher precision and recall than any other approaches. This result confirms that the POS, the occurrence position of the word and the length of the word are all the important factors which will affect the prediction of prosodic words.

Intermediate phrase is another bigger rhythmic unit that will affect the naturalness of TTS. We will concentrate on combining prosodic words into intermediate phrase in the future.

7. References

[1] Min Chu; Yao Qian, 2001. Locating boundaries for prosodic constituents in unrestricted Mandarin texts. Journal of Computational Linguistics and Chinese Language Processing 6(1), 61-82.

[2] Eric Sanders; Paul Taylor, 1995. Using statistical models to predict phrase boundaries for speech synthesis. In Eurospeech 95, Madrid , 1811-1814.

[3] Yao Qian; Min Chu, 2001. Segmenting unrestricted Chinese text into prosodic words instead of lexical words. In Proc. of ICASSP2001, Salt Lake City, 825 –828.

[4] Zhiwei Ying; Xiaohua Shi, 2001. An RNN-based algorithm to detect prosodic phrase for Chinese TTS. In Proc. of ICASSP2001, Salt Lake City, 809 –812.

[5] Brill E, 1994. A Rule-based Approach to Prepositional Phrase Attachment Disambiguation. In Proceedings of the 15th International Conference on Computational Linguistics, 1198-1204.