The Power of Special Characters in Prosodic Word Prediction for Chinese TTS

Zhengchen Zhang, Minghui Dong

Human Language Technology Department,
Institute for Infocomm Research (I2R), A*STAR, Singapore
zhangzc@i2r.a-star.edu.sg, mhdong@i2r.a-star.edu.sg

Abstract

Prosodic word (PW) prediction in Chinese Text-To-Speech (TTS) can be formulated as a classification problem that one predicts the tag of every character boundary in a sentence is the PW boundary or not. In this paper, a set of new features called special characters are introduced and put into classifiers to address the PW prediction problem. Some characters often appear at the beginning or at the end of a PW, which make them a strong clue of a PW boundary. Besides, quite a lot of PWs have only one character, which make such characters special. We select a set of special single characters, special starting characters, and special ending characters to help predict PW boundaries. Some special lexical words are often taken as PWs, and we collect a list of such words for PW boundary prediction. Decision tree, Supporting Vector Machine (SVM), MultiLayer Perceptron, and Random Forests are employed as the classifiers. Other features like part-of-speech (POS) of characters, word length, etc. are also used for PW prediction. In our experiments, we got 90.5% and 91.3% accuracies on two corpora containing $8,000$ and $1,349$ sentences respectively, which proved the efficiency of the method.

Index Terms: prosodic word prediction, speech synthesis

1. Introduction

The rhythm of Chinese speech is constituted by a hierarchical prosodic structure [1] that contains three levels: intonation phrase, prosodic phrase, and prosodic word. People tend to group words into small units and add short pauses between these units when they are speaking a long utterance to make their speech more easily understood. Prosodic word, which is in the lowest level of the hierarchical prosodic structure, is a set of syllables that are spoken continuously. In a Chinese TTS system, the prosodic word can be seen as a group of lexical words that are uttered together. For example, "姜还是老的辣" (The older the wiser) is often read as "姜|还是|老的辣" where "|" is a break mark, while the lexical words are "姜|还是|老|的|辣". In this paper, "还" and "是" are called characters. To generate natural speech by a TTS system, one must be able to detect the boundaries of the prosodic words (PWs). For a Chinese TTS system, the input is pure Chinese text like "姜还是老的辣". Normally, a Chinese word segmentation procedure is conducted to detect the lexical words first. Then different methods can be applied to predict the boundaries of PWs.

The PW prediction problem can be formulated as a classification problem that one predicts whether the boundary between two adjacent lexical words is a PW boundary or not. A statistical rule based method was proposed in [2] where the tag value T_1 means a boundary is a PW boundary, and T_0 denotes not. The method first categorized lexical word boundaries into different classes according to the word pairs before and after them. Part-of-speech (POS) and word length of the word pairs are features used to category the boundaries. Then the probability of being a PW boundary is estimated by $count(T_i|C)/count(C)$ where $count(C)$ is the total number of boundaries in a category C, and $count(T_i|C)$ is the frequency of boundaries labelled with tag T_i in C. A limitation of this method is that when the number of items in a category is small, the probability is not reliable. Statistical models [3] will always face this data sparseness problems.

Classification and regression tree (CART) was applied to solve the PW prediction problem in [2]. The CART tree approach is fast and it can achieve state-of-the-art system performance. However, this approach assumes that every word boundary is independent to each other, while it is not truth in the real world. To overcome this problem, a Markov Chain model was proposed in [4] to combine the CART approach and a statistical model. The probability of a word boundary being a PW boundary is determined by the features of current word and the previous boundary type. The previous boundary tag t_{i-1} and the features of current word Y_i are send into a CART tree to get the probability of the current boundary tag t_i. The probabilities of every word boundary in a sentence is calculated, and then a Viterbi search algorithm is run to maximize the probability of the whole sentence $\mathtt{P}(t_1, ..., t_n|Y_1, ..., Y_n)$. This method considered context information in PW prediction and achieved a good accuracy result, although it needs two steps to predict a boundary type.

Many works on prosodic phrase prediction can be good references for PW prediction problem because the tasks are similar and many useful features were introduced in these works [5, 6, 7]. The authors proposed many interesting features like phonetic information [5], text chunking features as well as syntactic tree features [6]. In [7], the lexical word sequences are used as a type of feature. Combined with POS, word position in a sentence, and word length features, 93.63% precision and 97.51% recall were obtained by a maximum entropy model for PW prediction. This is a very impressive performance. However, the computational load will be heavy as there are too many lexical word sequences. In this work, we only select some special lexical words that are taken as PWs directly. Instead of using the lexical word sequences, we just check whether a character is in a special lexical word or not.

The paper is organized as follows. Section 2 introduces the features used in this work. Experimental results are reported in Section 3. We conclude our work in Section 4.

2. Features for predicting PW

We will explain features used in this work using an example shown in Table 1 in this section. The features can be divided into six categories: Special Character, POS Sequence, Word Length Sequence, Special POS, Tone Sequence, and Word Position. The Special Character feature is described separately.

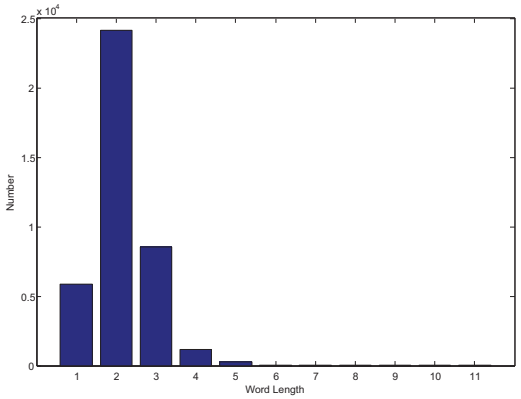
Features	Sample for word "是" in "姜 还是 老的辣"
Is Special Character	1
Is Special Starting	1
Is Special Ending	1
Is in Special Word (SW)	1
Position in SW (front to back)	1
Position in SW (backward)	0
POS sequence	nil nr v a n
Word Length sequence	-1 1 2 3 -1
Is Special Start POS	1
Is Special Middle POS	0
Is Special End POS	0
Special POS Length	2
Tone sequence	1 2 5 3 5
Word Position (front to back)	3
Word Position (backward)	4

Table 1: Features used for PW prediction.

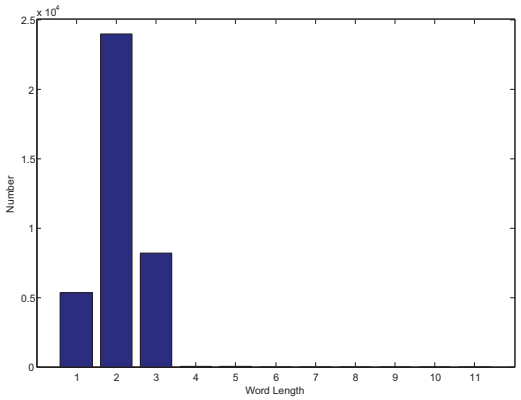
2.1. Special Characters

The special characters are selected from the training corpus. We introduce the corpora used in this work before describing the feature. Three corpora are used in this work. The first corpus, named Corpus 1, consists of $5,884$ sentences. The lexical word boundary, prosodic word boundary, pinyin including tones and POS are labelled manually. A total of $93,446$ character boundaries in it, and $40,030$ of them are PW boundaries. Corpus 2 contains $8,000$ sentences and $87,945$ character boundaries, in which $37,489$ are PW boundaries. In this corpus, the POS and the lexical word boundaries are not labelled manually. We use maximum matching (MM) method [8] to do the Chinese Word Segmentation (CWS). Usually, the accuracy of CWS obtained by the MM method is not very high, and many words are segmented into single characters due to the small size of the dictionary. However, this will not affect the PW prediction results because the most important features we used are character based. A basic statistical method without smoothing [9] is employed to tag the POS. Corpus 3 is composed of $1,349$ sentences that contains $19,551$ character boundaries with $8,319$ PW boundaries inside. Corpus 1 and Corpus 3 are labelled by same group of annotators. Corpus 2 is constructed by another laboratory.

We count the number of characters in a PW on Corpus 1 and Corpus 2. The statistical results are shown in Fig. 1. One can see that most of the PWs containing less than 4 characters. A total of $5,838$ prosodic words have only one character in Corpus 1, which contains 806 unique characters. Some characters are labelled as PWs frequently. For instance, "和" is labelled as a prosodic word for 639 times in our corpus. These special characters are often strong clues of prosodic word boundaries. We selected 42 such special single characters that appear more than 25 times in Corpus 1, and whether a character is in this list or not is taken as a feature. A total of $24,165$ PWs have two characters, and $8,544$ PWs contains three characters in Corpus 1. Given two consequence characters, if the first character often appears at the beginning of a PW and the second one normally appears at the end of a PW, this two characters may compose a PW. For PWs with three characters, it may be estimated using the same method. We count the frequencies of each character appearing at the special position of PWs. Take the character "的" as an example, it is labelled as the end of a PW for $4,239$ times in the corpus, and it appears for $4,400$ times in total. To employ such information in PW prediction, we selected 406 characters that appear at the beginning of a PW more than 25 times and 263 characters that appear at the end of a PW more than 25 times as our special character feature. From Fig. 1b, we can find similar trend exists in Corpus 2 in terms of character number of PWs. Some special characters are extracted



(a) Corpus 1



(b) Corpus 2

Figure 1: Prosodic word length in two corpora.

from Corpus 2 using the same method as above. Surprisingly, the results are highly coincident to those obtained from Corpus 1. Take the special single character as an example, a total of 76 and 78 characters that appearing more than 10 times were obtained from Corpus 1 and Corpus 2 respectively. There are 69 common characters in these two lists. The details about special starting characters and special ending characters with frequencies greater than 10 are shown in Table 2. This phenomenon demonstrates that the special characters for PW may be common in Chinese considering that there are about $5,000$ frequently used Chinese characters in total. The PW boundary may be enumerated in this case. Hence, we generate a special

Category	Corpus 1	Corpus 2	Common
Single	76	78	69
Starting	887	818	809
Ending	659	549	542

Table 2: Common special characters in two corpora.

single character list, a special starting character list and a special ending character list to compose features to train classifiers for PW detection. The features of a character are set to be 1 if it is in the lists. Otherwise, the feature values are 0. For example, the word "是" shown in Table 1 appears in all the three lists. Hence, the feature values are set to be $1, 1, 1$. The pre-previous word, previous word, next word and next-next word are also considered if they are available.

Prosodic words are composed by several lexical words usually. Some lexical words themselves are taken as prosodic words in the training set. We collect such words and count the times that they appear. We find that most of these words are proper nouns like country names, organization names, as well as technical terms etc., and most of them appear less than 5 times in the training set. It has a high probability that the boundaries after the last character of these words are PW boundaries. A list of $4,521$ special words is collected. Is a character in a special word is taken as a feature. The position of the character in a special word is also taken as a feature. As the last character is possible to be more important, the position in reverse order is considered.

2.2. Other features

Part-of-speech (POS) and length of a lexical word have been shown important for PW detection problems [4]. We consider the previous and following two words' information as well. Let x be current word, $POS_{x-2}, POS_{x-1}, POS_x, POS_{x+1}$, and POS_{x+2} are taken as features if they are available. Similarly, the lengths of previous and following words $WL_{x-2}, WL_{x-1}, WL_x, WL_{x+1}$, and WL_{x+2} are another kind of feature. There are a total of 30 POS categories in our system. Some of the categories contain very small amount of words. Too many POS categories with small frequencies may cause data sparseness problems [4]. We combine less frequent POS tags into one "*others*" category, and reduce the number of POS categories from 30 to 16. Some POS sequences appear very frequently in the corpus. For example, $6,617$ PWs have the POS pattern "n n", i.e., they are composed of two nouns. We collect 39 special frequent POS patterns into a list. If a POS sequence "v a" of word "还是|老" is in the list, the feature "Is Special Start POS" of characters "还" and "是" are 1. The feature "Is Special End POS" of the character "老" is 1. Some of the special patterns are embedded in other patterns. For example, both "n n" and "n n n" are special POS patterns. Another feature "Special POS Length" is added to differentiate the embedding cases from non-embedding ones. If a character in pattern "n n" and "n n n" at the same time, the "Special POS Length" is set to be $2 + 3 = 5$. In our experiments, the Special POS feature did not contribute much to the accuracy improvement. We still introduce this feature here because it presents a pattern hidden in the PW.

Tone of a word is also considered in this work. We find that the tone sequence is also a clue of a PW boundary. In our corpus, 4556 PWs have the tone sequence "4 4". Hence, the tones of a word and its neighbours are considered as features.

Word position in a sentence is a kind of feature that has been used in Prosodic Phrase prediction. Here we take the word position counting from the start of a sentence and the backward position as features. Combining with other features, they may be helpful for predicting the boundary types of the starting and ending words of a sentence.

3. Experimental Results

In this section, we report our experimental setup and the results obtained. We take Corpus 1 as the training set and Corpus 2 as well as Corpus 3 as the testing sets. We did not count the character at the end of a sentence, as it is always a PW boundary. Accuracy and F-measure are employed as the evaluation methods, which is calculated by

$$accuracy = N_c/N_a \tag{1}$$

where N_c is the correctly predicted boundary number, and N_a is the total number of boundaries.

$$Precision = \frac{tp}{tp + fp}, \tag{2}$$

$$Recall = \frac{tp}{tp + fn}, \tag{3}$$

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

where tp is the number of correctly predicted PW boundaries, and fp is the number of boundaries that are wrongly predicted as PW boundaries. fn is the number of PW boundaries that are not predicted.

We employ C4.5, supporting vector machines (SVM), MultiLayer Perceptron (MLP), and Random Forests (RF) as the classifiers. C4.5 is a well known CART tree construction method that has been used in many areas. Weka [10], which is an open source tool for machine learning, is employed in our experiments. The J48 classifier in Weka is an implement of C4.5 algorithm. We use it to evaluate the C4.5 algorithm. LibSVM [11] is employed to implement the SVM method. MLP and RF are implemented by Weka. All default parameters were used in our experiments unless otherwise stated. The system performance is reported in Table 3. One can find that C4.5 could obtain good results for both corpus. About 90.5% accuracy was achieved on Corpus 2, and the performance on Corpus 3 is 0.8% higher than that on Corpus 2. This may because that the Corpus 3 is annotated by the same team with the training corpus. SVM obtained best results on Corpus 2, while did not perform well on Corpus 3. Random Forests perform better on Corpus 3 than on Corpus 2. One of the disadvantages of SVM and MLP is that they consume much longer time than C4.5 and RF. To build a model on the training set, C4.5 takes about 6 seconds, while SVM needs 900 seconds and MLP needs one to two hours. The low dimensional feature vectors may be another reason for the fast training of C4.5 method. Most of the features are numbers, and the feature vector of each character has a total of 37 dimensions only.

To investigate the influence of each feature category, the system performance obtained by C4.5 with different feature sets are listed in Table 4. In this table, we can find that accuracies of 79.3% and 81.3% are achieved using the special character feature only. Even if we could not get very accurate Chinese Word Segmentation results, we still could get about 84.7% PW prediction accuracy on Corpus 2 with the special word feature. Starting from the second row, we add a new type of feature indicated by $+$. After adding the word position features, the

Corpus 2				
Method	Accuracy%	Pre	Rec	F1
SVM	90.9	0.906	0.877	0.892
C4.5	90.5	0.911	0.860	0.885
MLP	88.5	0.926	0.795	0.856
RF	89.2	0.904	0.836	0.869
Corpus 3				
Method	Accuracy%	Pre	Rec	F1
SVM	88.9	0.865	0.877	0.871
C4.5	91.3	0.901	0.895	0.898
MLP	89.6	0.914	0.834	0.872
RF	91.2	0.903	0.888	0.895

Table 3: System performance of different methods.

Corpus 2				
Feature	Accuracy%	Pre	Rec	F1
Special Character	79.3	0.777	0.722	0.748
+ Special Word	84.7	0.857	0.768	0.810
+ Word Position	86.5	0.869	0.805	0.836
+ Word Length	90.4	0.904	0.866	0.885
+ POS Sequence	90.7	0.914	0.864	0.888
+ Special POS	90.6	0.913	0.863	0.887
+ Tone	90.5	0.911	0.860	0.885
Corpus 3				
Feature	Accuracy%	Pre	Rec	F1
Special Character	81.3	0.792	0.761	0.776
+ Special Word	86.6	0.862	0.815	0.838
+ Word Position	88.1	0.871	0.846	0.858
+ Word Length	90.8	0.895	0.888	0.891
+ POS Sequence	91.2	0.901	0.890	0.896
+ Special POS	91.2	0.901	0.891	0.896
+ Tone	91.3	0.901	0.895	0.898

Table 4: Influence to system performance of different features on Corpus 2 and Corpus 3.

accuracy increased to 86.5% on Corpus 2. One can see that the accuracy increased about 3.9% for Corpus 2 after adding the Word Length feature. The POS sequence feature contributed about 0.3% of the increasing. Special POS did not help, and the performance decreased a little by contraries. The reason may be that the information has been contained by the POS sequence feature. Some special POS information mislead the classifier. For Corpus 2, the Tone information also decreased the accuracy. But it improved the performance a little on Corpus 3. The accuracy was 91.3% after adding the tone feature.

We cannot compare the results of our work to published results because different corpora were used in different papers. We list the reported results of some state-of-the-art methods in Table 5 as a reference.

Method	Accuracy%	Pre	Rec	F1
[4]	91.65	N.A.	N.A.	N.A.
[3]	N.A.	0.9183	0.8846	0.9011
[12]	N.A.	0.934	0.844	0.887
[7]	N.A.	0.9363	0.9751	N.A.

Table 5: System performance of reported methods.

4. Conclusion

In this paper, we have introduced special characters features to address the prosodic word prediction problem. Experimental results demonstrated that the special character features are very strong clues of prosodic word boundary. We have obtained about 80% accuracy in terms of accuracy by using special characters as features only. Experiments conducted on different corpora proved that the features can be used generally.

5. References

[1] L. Aijun, L. Maocan, C. Xiaxia, Z. Yiqing, S. Guohua, H. Wu *et al.*, "Speech corpus of chinese discourse and the phonetic research," in *ICSLP2000*, 2000.

[2] Y. Qian, M. Chu, and H. Peng, "Segmenting unrestricted chinese text into prosodic words instead of lexical words," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 825–828.

[3] H.-j. Peng, C.-c. Chen, C.-y. Tseng, and K.-j. Chen, "Predicting prosodic words from lexical words-a first step towards predicting prosody from text," in *Chinese Spoken Language Processing, 2004 International Symposium on*. IEEE, 2004, pp. 173–176.

[4] M. Dong, K.-T. Lua, and H. Li, "A probabilistic approach to prosodic word prediction for mandarin chinese tts," in *INTERSPEECH*, 2005, pp. 3245–3248.

[5] Z. Sheng, T. Jianhua, and C. Lianhong, "Learning rules for chinese prosodic phrase prediction," in *Proceedings of the first SIGHAN workshop on Chinese language processing-Volume 18*. Association for Computational Linguistics, 2002, pp. 1–7.

[6] Z. Sheng, T. Jianhua, and J. DanLing, "Chinese prosodic phrasing with extended features," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–492.

[7] F. Liu, H. Jia, and J. Tao, "A maximum entropy based hierarchical model for automatic prosodic boundary labeling in mandarin," in *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*. IEEE, 2008, pp. 1–4.

[8] P.-k. Wong and C. Chan, "Chinese word segmentation based on maximum matching and word binding force," in *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1996, pp. 200–203.

[9] T. Brants, "Tnt: a statistical part-of-speech tagger," in *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000, pp. 224–231.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[11] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[12] Z. Zhao and X. Ma, "Active learning for prediction of prosodic word boundaries in chinese tts using maximum entropy markov model," *Journal of Software*, vol. 8, no. 12, pp. 3222–3228, 2013.