SEGMENTING UNRESTRICTED CHINESE TEXT INTO PROSODIC WORDS INSTEAD OF LEXICAL WORDS

Yao Qian, Min Chu, Hu Peng*
Microsoft Research China *Shanghai Normal University[1]
yqian@shtu.edu.cn, minchu@microsoft.com

ABSTRACT

This paper stresses the importance of converting a string of lexical words to that of prosodic words in TTS systems by presenting the surface differences and perceptual differences between them. A statistical rule based method and a CART based method are proposed as solutions. Though ComplicatedSet based CART method performs the best, the achievement is obtained at the cost of heavy computation workloads needed by a parser. Statistical rule based method results higher recall but lower precision, comparing to SimpleSet CART method. It is very difficult to tell which is better, since we don't know which affects naturalness more, precision or recall. Both of them require only lexicon word segmentation and POS tagging in the preprocessing stage, and are easily realized in TTS systems. Results of the preference test discloses that significant improvements on naturalness are perceived when lexical word strings are converted into prosodic word strings by our approach.

1. INTRODUCTION

Since text in many Asia languages, such as Chinese, Japanese or Korean, has no visual cue for word boundaries, word segmentation becomes a basic requirement for almost all text analysis related efforts in these languages. Many works on word segmentation can be found[1][2]. However, segmenting a sentence into a string of *lexical words* (L-word) precisely is still far from enough for generating natural and beautiful prosody in TTS systems, since L-words do not always accord with the basic prosodic units in real speech. For example, in a Chinese sentence, "我买了一本好书 (I brought a good book)", each character itself is a L-word. Yet, in natural speech, the basic units of rhythm (or prosody) are "我", "买了", "一本" and "好书". For convenience, in this paper, the basic prosodic unit is referred to as *prosodic word* (P-word), which is defined as a group of syllables that are uttered closely and continuously in speech[3]. P-word is the lowest constituent in the prosodic hierarchy[4] and should have a perceivable prosodic boundary. In other words, no prosodic boundary can be perceived within a P-word and any level of pause should happen only on boundaries between P-words. A P-word can contain more than one L-word and it can also be only a part of a L-word. Many studies reveal that there are relationships between P-word and L-word[5][6], yet, not many works have been done on segmenting unrestricted text into strings of P-word. This paper solves the problem by data-driven methods. A statistical rule based approach and a CART based approach for converting strings of L-words into strings of P-words are presented and compared. A preference test is designed to investigate the perceptibility of differences between speech synthesized from L-word strings and P-word strings. The results are positive and encouraging.

Section 2 illustrates the importance of the conversion from a string of L-word to a string of P-word by presenting the surface differences between them. And the perceptual differences between P-word strings and L-word strings are presented in Section 5. A statistical rule based method and a CART based method for the conversion are described in Section 3 and 4. Results by different methods are compared and discussed in Section 6.

2. SURFACE DIFFERENCES BETWEEN P-WORD AND L-WORD

A large speech corpus, which contains 11248 sentences, has been collected and annotated. The length of these sentences is between 10 and 30 characters. P-word boundaries are annotated manually in the script of the corpus by perceptive tests. 1348 sentences are annotated three times by three annotators (HJY, ZF and ZR) separately and the resulted three annotations are compared in table 1, where precision and recall are given by

$$precision = CPWB / APWB * 100\% \quad (1)$$

$$recall = CPWB / ARPWB * 100\% \quad (2)$$

where, ARPWB, standing for *all real P-word boundary*, is the total number of real P-word boundaries. (If more than two annotators share the same opinion on the location of a boundary, the boundary is kept as a real one). APWB, standing for *annotated P-word boundary*, is the total number of P-word boundaries annotated by an annotator and CPWB (*correc P-word boundary*) is the number of boundaries annotated correctly by the annotator. From table 1, we find that very high ratio of agreement on the locations of P-word boundaries has been achieved among the three annotators. The remaining sentences are annotated only once by them to reduce workloads. A total of 77642 P-words are annotated. They are used as the ARPWB reference for all automatic methods presented in this paper.

Table 1. Precision and recall on P-word boundaries for three annotators.

Annotators	HJY	ZF	ZR
Precision (%)	98.9	98.5	99.3
Recall (%)	99.2	99.3	98.9

[1] Visiting student at Microsoft Research China

All sentences in the script for the speech corpus are segmented into L-words by a block-based robust dependency parser[7]. Totally 95831 L-words are obtained. This number is 23.4% larger than that of P-word. Comparing the L-word boundaries with ARPWB, we get 70.71% and 93.62% for the precision and recall respectively, which reveal the great differences between P-word and L-word. The distribution of length of P-words and L-words in the corpus is shown in Figure 1, from where we find that there are much more mono-character L-words than P-words and more bi-character P-words than L-words. The maximum length of P-word in the corpus is 5-character, while, the maximum length of L-word in the corpus is 13-character. A very important feature for P-word that discriminates it from L-word is that it is constrained not only by semantic requirement of a sentence, but also by the physical mechanism of articulators and the beauty of rhythm in speech. In order to meet the disyllabic rhythm of Mandarin, many mono-character L-words are uttered closely with their pre- or post- neighbors to form a P-word, and some long L-words that contain 4 or more characters are uttered as several short P-words in real speech.

If all L-words longer than 3 characters are splitted into several shorter P-words, the precision and recall rates increase to 71.69% and 98.8% respectively, which is used as the reference performance for our P-word segmentation methods. The splitting of longer L-word is realized by adding structural information into the lexicon. After performing the splitting, high enough recall is obtained, yet, the precision is far from satisfaction. A statistical rule based method and a CART based method are presented in this paper to increase precision. In all experiments, 76% of the corpus is used as training data and 24% of it is used as testing data.

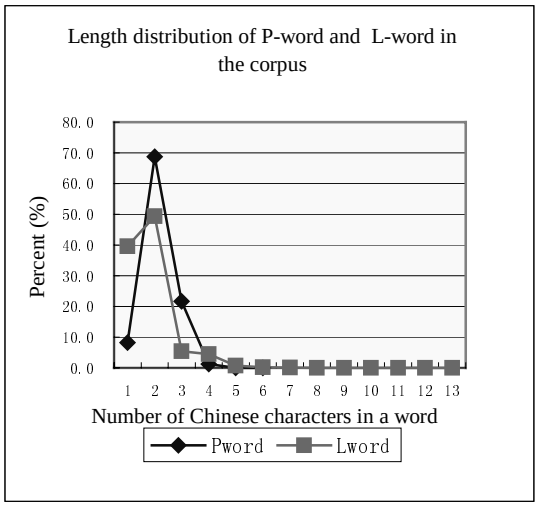


Figure 1. Length distribution of P-word and L-word in the corpus.

3. STATISTICAL RULE BASED METHOD

The boundary between two succeeding L-words is defined as a word juncture, which takes only two values in this paper. T_0 represents a word juncture that is not a P-word boundary and T_1 means a word juncture is a P-word boundary. Then, the problem of converting strings of L-words into strings of P-word becomes a problem of predicting the type of each word juncture. Part of speech (POS) of the words around a juncture is believed to be the most important cue for determining the type of the juncture. Other factors that may affect the type are the length of words, the juncture type before or after current juncture, and so on. Only POS and length of words are used in this section, and some phrase level information is used in Section 4.

Lexical words are classified into categories by their features such as the Part-Of-Speech and the length of words. A word juncture is represented by the pair of categories of the L-words before and after it, which is denoted as $P_i(i=1,2,...,I)$, where I is the total number of category pairs appeared in the training data. $count(P_i)$ counts the times of a P_i appeared in the training set. And the number of times of all word junctures with P_i take type T_0 is denoted as $count(T_0/P_i)$. The conditional probability for a juncture with P_i taking value T_0 can be estimated by:

$$\tilde{P}(T_0/P_i) = count(T_0/P_i) \Big/ count(P_i) \qquad (3)$$

$\tilde{P}(T_0/P_i)$ is the probability for two concatenated L-words with P_i to be merged into one P-word. When $count(P_i)$ is a small number, the estimated probability are not reliable. A *weighted probability* (WP), given in formula (4), is adopted to reduce the contribution of not reliable probabilities, while keeping the contribution of the reliable ones.

$$W\tilde{P}(T_0/P_i) = \tilde{P}(T_0/P_i) * W(P_i) \qquad (4)$$

$W(P_i)$ in equation (4) is a function taking values between [0,1] and increasing monotonously with $count(P_i)$. The sigmoid function is used to construct the weight function in our approach:

$$W(P_i) = sigmoid(1+\log(count(P_i))) \quad count(P_i) > 0 \qquad (5)$$

For deciding whether two concatenated L-word should be merged, a threshold of the WP, θ, is used. All junctures with P_i, which satisfy equation (6), are assigned to type T_0.

$$W\tilde{P}(T_0/P_i) >= \theta \qquad (6)$$

The others are assigned T_1. Then, all P_i, that satisfy equation (6) are rules for combining neighbored L-words. When θ decreases from 1 to 0, the number of rules, denoted as Rsize, increases from none to I. Precision and recall changed with θ too.

Experiments are done for three category pair sets with θ changing from to 1.0 to 0. The three sets are:

- **Simple POS set (SPS)**: 17 categories of POS are used and 270 POS pairs are found in the training set.
- **Word length indicated POS set (WLIPS)**: each simple POS category splits into several by indicating the length of L-words. For example N1, N2, N3 represented mono-, bi-,

and tri-character noun respectively. Total of 602 POS pairs are found in training set.

- **Extended WLIPS (EWLIPS)**: 100 frequently used mono-character words are treated as special POS categories and they are added with WLIPS to form the EWLPS. Totally 2739 POS pairs are found in training set.

Precisions and recalls for each set are listed in table 2. And their first order differential contours are shown in figure 2. The best results in the three sets are those at the points where differential of both precision and recall is very small. In our case, best results are obtained when $\theta = 0.7$, $\theta = 0.6$ and $\theta = 0.5$ in SPS, WLIPS and EWLIPS respectively. All of them increase precision at the cost of decreasing recall to some extents. Extending SPS to WLIPS brings in 6.46 percent increasing on precision at the cost of 2.27 percent decreasing on recall. We think WLIPS performs better. When extending WLIPS to EWLIPS, only slightly changes in precision and recall are found. Yet the Rsize increases from 283 to 2022. We think it is unworthy.

4. CART BASED METHOD

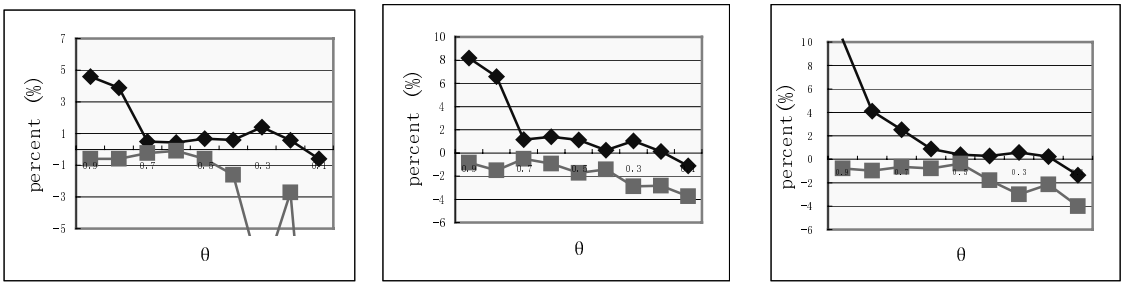
The problem described in Section 3 can be treated as an automatic learning problem. Automatic *classification and regression tree* (CART) is used to solve the problem. Two question sets have been used. One of them, denoted as SimpleSet, contains only questions about the POS and length of the concatenated L-words pair, in which amount of information used is the same as that in WLIPS in Section 3. Additional information, such as whether current word juncture is a phrase boundary, the length of current phrase, and the type of previous juncture, is questioned in another question set, denoted ComplicatedSet. Only simple questions are listed in both question sets Composite-questions will be automatically constructed during the growing phase of the tree to avoid the over-fragmented problem[8]. The resulted precisions and recalls for testing data are listed in table 3. From Table 3, we found that those additional questions do helps on predicting the type of a juncture. However, a syntactic parser is needed for obtaining those phrasal level information and the computational workload requested by a parser is often heavy. Comparing with results in table 2, we find that CART based method results higher precision but lower recall. More discussions are presented in Section 6.

Table 3. Precisions and recalls of P-word boundaries generated by CART trained from two question set, SimpleSet and ComlicatedSet.

Q set	Precision (%)	Recall (%)
SimpleSet CART	90.79	91.52
ComplicatedSet CART	92.41	94.48

Table 2. Precisions, recalls and Rsizes in different category pair set with regarding to different threshold θ.

	SPS			WLIPS			EWLIPS		
θ	Rsize	Precision(%)	Recall(%)	Rsize	Precision(%)	Recall(%)	Rsize	Precision(%)	Recall(%)
1	0	71.69	98.80	0	71.69	98.80	0	71.69	98.80
0.9	27	74.99	98.20	56	77.55	97.97	498	78.99	98.03
0.8	44	77.90	97.62	114	82.65	96.50	863	82.22	97.08
0.7	112	78.29	97.40	258	83.59	96.02	1827	84.28	96.45
0.6	129	78.63	97.31	283	84.76	95.13	1964	85.03	95.68
0.5	136	79.16	96.72	304	85.71	93.50	2022	85.35	95.32
0.4	168	79.64	95.17	358	85.91	92.16	2229	85.58	93.62
0.3	197	80.75	87.28	405	86.81	89.51	2385	86.07	90.82
0.2	216	81.21	84.90	439	86.92	86.99	2488	86.26	88.86
0.1	243	80.72	65.48	502	85.95	83.73	2618	85.09	85.29
0	270	75.03	53.63	602	77.90	61.52	2739	78.90	65.27



(a) (b) (c)

Figure 2. First order differential contours of precision and recall in three category pair sets. (a) in SPS; (b) in WLIPS; (c) in EWLIPS
◆ represent precision ■represent recall

5. PERCEPTUAL DIFFERENCES BETWEEN P-WORD STRINGS AND L-WORD STRINGS

A perceptive experiment is designed to investigate the perceptual differences between P-word strings and L-word strings. Speech waves are synthesized with our data-driven TTS system[9], which takes in three types of inputs:

TypeA: sentences with manually annotated P-word boundaries.

TypeB: sentences with P-word boundaries generated automatically by ComplicatedSet based CART method.

TypeC: sentences with L-word boundary only.

Three-version speech waves of total 108 sentences picked up from the testing set are synthesized. And 6 comparing pairs (AB, BA, BC, CB, AC and CA) are formed for each sentence. Totally 15 subjects take part in the experiments, each of them listens to part of these comparing pairs and is forced to select a better utterance in each pair. The preference rate is counted as:

$$P_T = count(T) / \sum count(T), \quad T=A, B, or\ C \qquad (7)$$

where, P_T is the total number of times when type T is selected.

The final preference rates for all three types are shown in figure 3. It can be found that typeA (manually annotated P-word strings) has the highest preference rate. Only slightly differences can be perceived between typeA and typeB (the automatically generated P-word strings). And, typeB sounds much better than typeC (L-word strings). This result elucidates the importance of converting L-word strings to P-word strings from the perceptual point of view and it also shows that our automatic method performs closely to the manually annotations.

6. DISCUSSION AND CONCLUSION

This paper addresses the problem of P-word segmentation by presenting the differences between P-words and L-words on both surface and perceptive aspects. A statistical rule based method and a CART based method are proposed. The ComplicatedSet based CART method achieves both higher precision and recall at the cost of using a syntactic parser for the additional phrase level information. Statistical rules based on WLIPS result higher recall but lower precision, comparing to SimpleSet CART method. Though EWLIPS based rules increase precision slightly, it is unworthy to increase the size of rule set from 283 to 2022. It is very difficult to tell which is better between WLIPS based rules and SimpleSet based CART, since we don't know which is more important for naturalness, precision or recall. Both methods require only L-word segmentation and POS tagging as preprocessing and are easily realized in TTS systems. Though the problem of converting L-word strings to P-words strings are illustrated in Chinese in this paper, similar issues are faced in many other Asia languages like Japanese and Korean. We will extend our approach for these languages in the future.

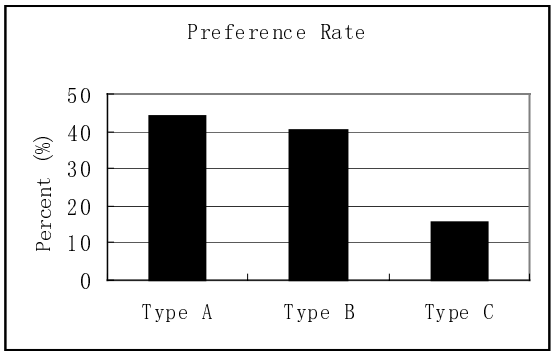


Figure 3. Preference rates for three types of synthesized speech. Type A, synthesized from manually annotated P-word strings; Type B, synthesized from automatically annotated P-word strings; Type C, synthesized from L-word strings

7. ACKNOWLEDGMENTS

The authors are especially grateful to Ming Zhou for providing the block-based robust dependency parser as a baseline tool and for his helps on using it. The authors thank everybody who takes part into the perceptual test.

8. REFERENCES

[1] Huang,X., Luo Z. and Tang, J, "A quick method for Chinese word segmentation", Intelligent Processing Systems, 1997, Vol.2, 1773-1776

[2] Wong, P. and Chan, C., "Chinese word segmentation based on maximum matching and word binding force", COLING'96, Copenhagen

[3] Chu, M. "Research on Chinese TTS system with high intelligibility and naturalness", PhD thesis at Institute of Acoustics, CASS, 1995.

[4] Nespor, M. and Vogel, I., Prosodic Phonology, Foris Publications, Dordrecht, Holland, 1986.

[5] Selkirk, E., Phonology and Syntax: The Relation between Sound and Structure, Cambridge, Mass.: MIT Press, 1984.

[6] Kaisse, E., Connected Speech: The Interaction of Syntax and Phonology, Academic Press, New York, 1985.

[7] Zhou, M., "A block-based robust dependency parser fir unrestricted Chinese text", The second Chinese Language Processing Workshop attached to ACL2000, Hong Kong, 2000

[8] Huang, X.D., Acero, A., Hon, H. and Meredith, S., Spoken Language Processing(draft), chapter 4

[9] Chu, M, Peng, H., Yang, H. and Chang, E., "Selection non-uniform units from a very large corpus for concatenative speech synthesizer", another paper submitted to ICASSP2001.