Journal of Computer Science 2 (8): 660-664, 2006
ISSN 1549-3636

Comparison of Approaches for Predicting Break Indices in Mandarin Speech Synthesis

Shao Yan-qiu, Zhao Yong-zhen, Han Ji-qing and Liu Ting
Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China

Abstract: This study adopts a large-scale corpus with five-tier break indices annotated according to C-TOBI. Based on it, several approaches, N-gram, Markov model and decision tree learning are applied to predict break indices automatically for unrestricted mandarin text. These approaches differ mutually not only in model, but also on features and even part-of-speech tag size. A deep comparison and analysis among these approaches was made in the research.

Key words: Markov models, speech synthesis, break indices, n-gram, decision tree

INTRODUCTION

One of the challenges of speech synthesis systems is to generate very natural and expressive synthesized speech which needs appropriate prosodic parameters. Many current synthesizers produce prosody in two steps. First, prosodic events are predicted at the symbolic level, which involves specifying break indices and pitch accent. Second, this symbolic representation receives a phonetic realization in terms of F0 contour, duration and volume. This paper deals with assigning break indices automatically for unrestricted mandarin text on the symbolic level.

A number of approaches have been proposed for such a task, ranging from simple to complex. In the earlier study, rules were written to locate prosodic boundaries. For instance, Bachenko and Fitzpatrick[1] built computational grammar using information about syntactic constituency, adjacency to a verb and constituent length to determine prosodic phrasing for synthetic speech. As it is known that rule driven method is slow, costly and inflexible which needs the rule writer to have all-round and deep understanding of the prosodic structure of this language. However, with the improvement of the computer processing ability, the large scale corpus becomes popular and stochastic statistical models have been applied more frequently for the advantage of automatic training and easily being planted to other domains. For example, CART was applied by Hirsburgh[2] to predict break indices using features such as punctuation, par of speech (POS), pitch accent types . Alan black and Paul Taylor[3] applied Markov model to assign phrase breaks from POS sequences. Other more complex stochastic methods have been tried by Ostendorf and Veilleux[4] who proposed a hierarchical stochastic model. These publications mentioned above, however, are all focused on English which is different from mandarin in nature. There is also relative research on this task for mandarin. In some works by Chu[5] in MSAR, break indices have been predicted using CART from the information such as POS, the distance from the beginning or the end of a sentence, the length of the sentence etc. Tao[6] also tried the same model, but using not only the features that can be abstracted from text but also the acoustic features and achieved perfect performance. Moreover, because of the particularity of mandarin itself, some work has been done using the special word class in Mandarin of empty word and auxiliary word in a sentence to predict the boundary and its type.

As it can be seen from the previous works, different information can be used to help perform this task, such as POS, phrase length, pitch accent, syntactic structure, acoustic features and so on. However, an important restriction is that our work needs to be integrated to the real speech synthesis systems and the features we can apply are only those that can be easily and reliably extracted from the raw text relying on some efficient text analyzer. Pitch accent itself is tougher to predict than break indices. Current automatic syntactic and semantic analyzers produce such poor performance that they can not be applied on this task. Moreover, there are no acoustic features that can be used for arbitrary input texts without corresponding speech. So the most frequently used features for this task are POS and word length.

In addition to the features used, there are other factors affecting the performance of the automatic assignment, such as the model and even the tag size of the POS set. To have a full comprehension of the effect of these factors on the automatic prediction of break indices, this paper will make a full-scale comparison from these three aspects and give a deep description of the task.

Corpus and word segmentation and POS tagging: Statistical models need training data to learn from it and in this paper a large scale corpus containing 12,000 sentences is adopted. Each sentence has its corresponding utterance, so the break indices were annotated manually by experienced annotators according to both the script and the relative speech.

Speaking of the annotation of break indices (BI),

Corresponding Author: Shao Yan-qiu, Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China

it is necessary to mention the prosodic constituents. There are many reports specifying various hierarchical structures for prosodic constituents. TOBI[7] is a proposed standard for transcribing symbolic prosody of American English utterances, which can be adapted to other languages as well with some modification. C-TOBI[8] is such a standard for mandarin speech synthesis which was proposed by the Phonetic Laboratory of the Institute of Linguistics, Chinese Academy of Social Science. On the break index tier, the prosodic association of words in an utterance is shown by labeling the end of each word for the subjective strength of its association with the next word, on the scale 0 to 4 which are abbreviated as BI0, BI1, BI2, BI3, BI4 for convenience. Here are the concrete definitions of them. BI0: the minimum break between syllables, usually breaks within a prosodic word; BI1: prosodic word boundary; BI2: minor prosodic phrase boundary; BI3: major prosodic phrase boundary; BI4: prosodic group boundary.

Unlike English, there is no blank between words in mandarin. So word segmentation is the fundamental step before almost any kind of text analysis and processing. Because this module connects closely with POS tagging, for mandarin these two modules are always integrated to be one system. As for the size of the POS set, there are various classifications with different granularity. In this paper, to test the influence of tag size on the performance of this task, three different word segmentation and part of speech tagging systems are applied. For convenience, they are called as System1, System2 and System3 accordingly. System1's tag size is 58 where every type of punctuation is given a single POS tag. And System2 has the tag size of 28 without specification for every type of punctuation. System3 is System2 by expanding its tag size and regarding every type of punctuation as different, so System3's tag size is 37

Approaches: Firstly, we formally define the problem as follows. Each character in the sentence is assumed to be followed by a boundary site (BS) and the break indices are supposed to label the types of every BS. After word segmentation and POS tagging, we get a series of lexical words. For each BS within a lexical word, we cannot predict its BI using the information of POS and it is assumed to have the type BI0. In fact, some rules should be written to locate the BI within the lexical word, but since it is not the consequential part of this paper, emphasis will not be put on it. Between every pair of words there is a juncture, which can take one of the five break indices. In the case of this paper the set of break indices consists 0 to 4. Then the task is changed to choose the most proper BI for each juncture. To complete this task, several approaches are proposed which are different from each other by models, information adopted and word segmentation and POS tagging systems which will be described in detail as following.

N-gram model: The simplest approach for assigning BI is to give every juncture the type with the largest possibility. In this model, the POS of the surrounding words of the juncture are used. Assume that the sentence S to be annotated contains L words after word segmentation and the POS tagging and the POS sequence of these L words is $c_1,c_2\ldots\ldots c_L$. M words before the juncture and N words after the juncture is adopted to be the context of the juncture, namely the window size is $M+N$. Then the task can be defined by equation (1).

$$\arg\max_{j_i} P(j_i \mid c_{i-M+1,\ldots\ldots} c_{i+N}) = \arg\max_{j_i} \frac{T(j_i, c_{i-M+1,\ldots\ldots} c_{i+N})}{T(c_{i-M+1,\ldots\ldots} c_{i+N})} \quad (1)$$

Where j_i means the juncture between word c_i and word c_{i+1} .The parameter $P(j_i \mid c_{i-M+1}\ldots c_{i+N})$ can be estimated from the training data using maximum likelihood estimation. Here $T(j_i, c_{i-M+1}\ldots c_{i+N})$ represents the occurrence times of sequence $c_{i-M+1}\ldots j_i \ldots c_{i+N}$ and $T(c_{i-M+1}\ldots c_{i+N})$ is the occurrence time of the POS sequence $c_{i-M+1}\ldots c_{i+N}$ in the corpus.

Markov model: This task can also be seen as a problem of sequences tagging on which Markov Model (MM) works well. MM model considers not only the emission probability of an observational output on a state, but also the transitional probability from one state to another. Thus, more contextual information could be used. For the problem of BI annotation, the observation sequence is the POS sequence $c_1c_2\ldots c_L$ and the state sequence is a BI sequence $j_1j_2\ldots j_{L-1}$ ($j_i \in \{0\text{-}4\}$). This can be seen as a five-state Markov chain. Thus, the problem is converted into finding a best state sequence $j_1j_2\ldots j_{L-1}$ to obtain the maximum probability of $P(j_1,j_2\ldots_{L-1}|c_1,c_2\ldots c_L)$.Here, equation (2) is employed.

$$\begin{aligned} &\arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} P(j_1 \ldots j_{L-1} \mid c_{1\ldots} c_L) = \\ &\arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} \frac{P(j_1 \ldots j_{L-1}) P(c_{1\ldots} c_L \mid j_1 \ldots j_{L-1})}{P(c_{1\ldots} c_L)} \end{aligned} \quad (2)$$

For the same POS sequence, the denominator of equation (2) is the same, so it can be neglected. Furthermore, the MM to be adopted here is the first order MM model, i.e. the transition probability is only related with the only one former state and the observation value is only related with the current state. So, equation (3) and (4) is got.

$$\begin{aligned} &P(j_1, j_2 \ldots j_{L-1}) = P(j_1 \mid j_0) \\ &P(j_2 \mid j_1) \ldots P(j_{L-1} \mid j_{L-2}) \end{aligned} \quad (3)$$

$$\begin{aligned} &P(c_1, c_2 \ldots c_L \mid j_1, j_2 \ldots j_{L-1}) \\ &= P(c_1, c_2 \mid j_1) P(c_2, c_3 \mid j_2) \ldots P(c_{L-1}, c_L \mid j_{L-1}) \end{aligned} \quad (4)$$

According to equation (3) and (4), equation (2) can be simplified as (5). All the parameters may be obtained from training data through statistical method and Viterbi algorithm is used to get the best state sequence.

$$\begin{aligned} &\arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} P(j_1 j_2 \ldots j_{L-1} \mid c_1 c_2 \ldots c_L) \\ &= \arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} \prod_{i=1}^{L-1} P(j_i \mid j_{i-1}) P(c_i c_{i+1} \mid j_i) \end{aligned} \quad (5)$$

Both the transition probability and the emission probability could be got by maximum probability estimation in the training corpus.

If the information of word length is taken into

consideration, the observation sequence will become c_1, l_1,c_2, l_1,…c_L l_L. Then the problem will be solved by finding a best state sequence to obtain the maximum probability of $P(j_1, j_2…j_{L-1}| c_1, l_1,c_2, l_2,…c_L l_L)$. The equation is given bellow.

$$\begin{aligned}&\arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} P(j_1 j_2 \ldots j_{L-1} \mid c_1 c_2 \ldots c_L) \\ &= \arg\max_{j_1 j_2 \ldots\ldots j_{L-1}} \prod_{i=1}^{L-1} P(j_i \mid j_{i-1}) P(c_i c_{i+1} l_i \mid j_i)\end{aligned} \tag{6}$$

Decision tree learning: Decision tree learning is a widely used algorithm for approximating discrete-valued target function. Of the family of decision tree learning, C4.5 is the most popular which is adopted in this paper.

Decision tree learning method can produce the tree by automatic feature selection by means of information entropy. Therefore, it's input are discrete valued candidate feature. In this experiment, the feature set for classification includes the POS and length of the *M* words before and *N* words after the juncture. Moreover, there is pruning procedure in C4.5 to avoid the problem of over fitting. Therefore, some data must be separated from the training set for validation.

Evaluation criteria and results: As mentioned above, this paper adopted a large scaled corpus containing 12,000 sentences, of which 9,000 sentences are used for training and 3,000 are used as test set. With respect to the problem of evaluation of the performance, accuracy is the traditionally used criteria for tagging problem, so overall accuracy for all BS was calculated using equation (7); then precision and recall were calculated for each BI type separately which are defined by equation (8) and equation (9).

$$Accu = C(B_p)/C(B) \tag{7}$$

$$\Pr e_i = C(B_{pi})/C(B_i) \tag{8}$$

$$\mathrm{Re}\, c_i = C(B_{pi})/C(B_{ri}) \tag{9}$$

Where $i \in \{0,1,2,3,4\}$ denotes the type of BI. $C(B)$ is the total number of BI in the test. Since every character is followed by a BI, $C(B)$ is also the total number of the characters in the test set. $C(B_p)$ is the number of the correctly predicted BS. $C(B_i)$ denotes the number of BS annotated as BI type i. $C(B_{pi})$ represents the number of annotation correctly predicted as the type *i*. $C(B_{ri})$ is the number of annotation in the test set with the type i.

However, the above evaluation criteria are a little coarse grained because it regards all the annotating error as the same. In fact, different types of errors will affect the synthesized result to different extents. For example, if a juncture of BI0 is wrongly annotated as BI4 or BI1, it's evident that the error of annotating BI4 will destroy the result more fiercely while BI1 is more acceptable by contrast. To have a more fine-grained evaluation of the performance, the criteria of Error Cost was proposed by Chu[5] in MSRA which is defined by equation (10).

$$ErrCost = \sum W_i C(E_i) \tag{10}$$

Where $C(E_i)$ denotes the number of BI errors equaling i which is defined as the difference between the assigned BI and the real one. W_i represents the weight for the error E_i. Evidently, Error Cost is the function of the size of corpus where the larger the size of corpus, the larger the Error Cost. To avoid this dependency and facilitate the comparison between the results tested on different corpus, Average Error Cost is defined which means the average value of Error Cost on all BS.

$$AverErrCost = \sum W_i C(E_i) / C(B) \tag{11}$$

In our case, there are four types of errors: E_1, E_2 ,E_3, E_4 and we specified that W_1=0.5, W_2=1, W_3=2, W_4=4.

As mentioned above, the test set contains 3,000 sentences and includes 48677 BS. These approaches are tested on it with varied control parameters. Here gives the parameter control for every approach.

Firstly, the result of N-gram model, since there is no prior knowledge on how large the window size will produce the best performance, so unigram, bigram and trigram of POS was applied and the results are listed in Table 1.

Then Table 2 gives the results of the basic Markov model and the Markov model with word length. Both of them are applied to System1.

C4.5 algorithms are applied to not only the corpus processed by system1 but also the corpus processed by System2 and System3 and these results can be seen in Table 3. Generally speaking, to facilitate the comparison of the models, all the algorithms are applied to System1. And then, for convenience, only the C4.5 algorithm was selected to work on System2 and System3 to see the influence of tag size.

Comparison of the results: The results above validate our assumption that the model, the information adopted and the size of POS set will affect the performance of automatic assignment of BI for mandarin text to different extents. And a comparison can be made between these approaches.

Of all the approaches above, N-gram Model is the simplest but quite effective, whose performance can be considered as the baseline of all the experiments. From the Table 1, it can be seen that the added information can help improve the result such as from unigram to bigram and trigram, but bigram model got the best performance. That's because with the increasing of the

Table 1: The result of N-gram model

Model	Accu	AverErrCost	P&R	BI0	BI1	BI2	BI3	BI4
Unigram+	66.8%	0.307862	Pre	66.8%	51.5%	39.9%	69.3%	93.1%
System1			Rec	96.4%	15.8%	18.8%	16.2%	90.7%
Bigram+	74.1%	0.19095	Pre	81.7%	52.5%	49.4%	60.0%	92.5%
System1			Rec	93.5%	45.6%	42.4%	22.3%	91.4%
Trigram+	74.1%	0.211732	Pre	80.4%	54.8%	51.0%	57.7%	92.7%
System1			Rec	94.0%	44.1%	41.8%	28.0%	89.0

Table 2: The result of MM

Model	Accu	AverErrCost	P&R	BI0	BI1	BI2	BI3	BI4
MM+	75.6%	0.164297	Pre	85.2%	54.2%	51.9%	53.4%	92.2%
system1			Rec	91.9%	52.1%	45.6%	35.8%	91.6%
MM+WordLen	77.0%	0.154755	Pre	86.7%	56.6%	54.6%	52.2%	92.2%
+system1			Rec	92.9%	53.2%	49.4%	37.8%	91.4%

Table 3: The result of decision tree learning

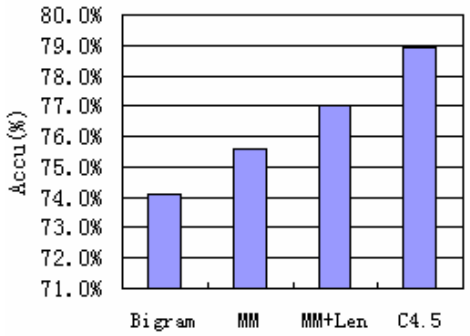
Model	Accu	AverErrCost	P&R	BI0	BI1	BI2	BI3	BI4
C4.5+	78.9%	0.150364	Pre	89.0%	59.8%	54.3%	51.4%	91.4%
System1			Rec	96.4%	57.9%	48.1%	29.5%	91.4%
C4.5+	78.5%	0.174504	Pre	90.0%	60.4%	54.9%	43.5%	84.2%
System2			Rec	95.6%	59.0%	51.3%	20.9%	90.1%
C4.5+	79.9%	0.139522	Pre	90.9%	60.3%	55.8%	50.9%	91.4%
System3			Rec	96.8%	58.1%	52.8%	30.5%	92.1%

dimension of the statistical information, the problem of data sparseness will become more and more serious which will outweigh the benefit that added information can bring out. Moreover, more information may mean more noise which will also damage the performance. So when we applied MM, only bigram is implemented and the result of bigram model will be regarded as the baseline.

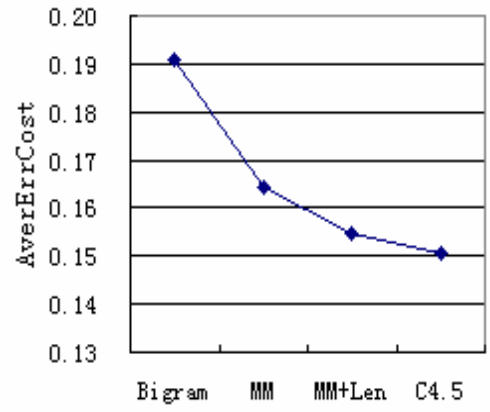
MM is the most widely used method for sequential tagging and as for this problem, the basic MM receives 1.5% increase than baseline in overall accuracy, but the Average Error Cost was decreased 14.2% which means the considering of interrelation between the break indices can help avoiding of great errors. What's more, after the introduction of word length, compared with the basic MM, the accuracy was increased 1.4% and the Average Error Cost was reduced 5.8% which proves that word length is also of great importance for this problem.

Speaking of Decision Tree learning, it is the method containing the best result than other approaches. Decision tree is a method which induces using statistical manner but its representation is actually rule, so it contains the advantages of the two kinds of methods which are the effectiveness and easy planting of statistical method and the flexibility of rules. Therefore it is more effective than the others. Compared with the approach of MM with word length, the accuracy was increased 1.9% and the average error cost was reduced 3%. Furthermore, it is quite quick and easy than other two methods on the implementation.

The comparison of these approaches using System1 can be seen clearly from Fig. 1a is the accuracy of these models and (b) is the AverErrCost of these models.



(a) The accuracy of different models



(b) The AverErrCost of different models

Fig. 1: The comparison of these approaches

On C4.5 algorithm, we also tested another factor affecting the performance of the task, namely POS tag size. The accuracy using System3 is 1.2% higher than using System1 and the Average Error Cost is 7.2% lower. The result shows that the size of the POS set also affects the outcome of the automatic assignment of break indices greatly. The classification of POS is more fine-grained, the POS tagging system's performance is less accurate and the data is more sparser. But punctuation is different from other POS which load decisive information for the categorization and the more specified, the better the performance. This conclusion can be made from the comparison of the results of the C4.5 algorithm using System2 and System3. So for the task of BI assignment, we should limit the tag size but give each type of punctuation a unique POS.

DISCUSSION

The problem of assigning BI is a complex weave of feature, algorithm and POS tag size. From experiments made above, we can conclude that to some extent the more information is used, the better the result will be obtained. But it depends on the choice of features and the result will be damaged if many kinds of information are just accumulated together without choice. Because of the problem of data sparseness and inflexibility in utilization of information, complete statistical method such as MM is not very proper for this task, while the method of decision tree, which has the representation of rule and deduction through statistical data, can work very well on this task. And for the size of POS set, appropriate tag size should be used and too fine-grained classification will reduce the result, but the punctuation is quite useful for the annotating and should be applied separately.

Considering the factors which will damage the performance, there are some reasons beyond the approaches. Firstly, the automatic word segmentation and POS tagging system cannot get 100% accuracy and the errors will be transferred to the task and damage the performance of the approaches greatly since we are leaning heavily on the information of POS and word length. Secondly is the problem of data sparseness which is inevitable for any statistical method. Thirdly it is the complexity of the task itself which may be determined not only by basically simple information such as POS and word length, but also the information about syntactic structure, semantic and even phonological information. For example, to balance the whole rhythm in the speech of the sentence, break will be inserted at the position which cannot be predicted by just the local POS and word length.

Last but not the least, concerning the comparison between different approaches, we focus on the criteria of accuracy and Average Error Cost which are synthesized evaluations and if we come to look at the precision and recall for each single break index, we see that BI0 and BI4 got much higher results than other tiers. That is because the confusion set size of these two scales is smaller. For example, for BI0, it can be mistook as BI1, but rarely as BI2, BI3 and BI4 whose behaviors are more different from BI0 and cannot be easily confused. But for BI1, it can be mistaken as BI0 and BI2 mostly. Generally speaking, the scale is always confused with the scales nearly neighbored. Moreover, BI0 can be good predicted by POS and word length and BI4 by punctuation. But BI1, BI2 and BI3 concern more with the structure of the sentence and balance of phonology information which are not available in our experiment.

Future work: Since sequential tagging algorithm can get lower Average Error Cost and decision tree learning can help improve the accuracy, we will consider a synthesized approach to integrate the advantages of these two methods. Moreover, the tagging results for BI1, BI2, BI3 are not very good, we'd like to build separate model and consider more information to process the tagging of these indices.

REFERENCES

1. Bachenko, J. and E. Fitzpatrick, 1990. A computational grammar of discourse neutral prosodic phrasing in English. Computational Linguistics, 16: 155-170.
2. Hirschberg, J. and P. Prieto, 1996. Training intonational phrasing rules automatically for English and Spanish Text-to-speech. Speech Communication, 18: 281-290.
3. Alan, W.B. and P. Taylor, 1998. Assigning phrase breaks from part-of-speech sequences. Computer Speech and Language, 12: 99-117.
4. Ostendorf, M. and N. Veilleux, 1994. A hierarchical stochastic model for automatic prediction of prosodic boundary location. Computational Linguistics, 20: 27-54.
5. MinChu, Y.Q., 2001. Locating boundaries for prosodic constituents in unrestricted Mandarin texts. Computational Linguistics and Chinese Language. Processing, 16: 1-22.
6. Tao, J., 2004. Acoustic and linguistic information based Chinese prosodic boundary labeling. Proc. Intl. Symp. Tonal Aspects of Languages, pp: 181-184
7. ToBI Intonation Transcription Summary. http://www.cs.indiana.edu/~port/teach/306/tobi.summary.html.
8. C-ToBI: Prosodic labeling system for Chinese. http://www.cass.net.cn/chinese/s18_yys/yuyin/product/product_10.htm.