BREAK INDEX LABELING OF MANDARIN TEXT VIA SYNTACTIC-TO-PROSODIC TREE MAPPING

Xiaotian Zhang[1,2,3] *Yao Qian*[3] *Hai Zhao*[1,2] *Frank K. Soong*[3]

[1]Center for Brain-Like Computing and Machine Intelligence,
Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University, Shanghai, China
[3]Microsoft Research Asia, Beijing, China
xtian.zh@gmail.com, yaoqian@microsoft.com,
zhaohai@cs.sjtu.edu.cn, frankkps@microsoft.com

ABSTRACT

In this study, we investigate the break index labeling problem with a syntactic-to-prosodic structure conversion. The statistical relationship between the mapped syntactic tree structure and prosodic tree structure of sentences in the training set is used to generate a Synchronous Tree Substitution Grammar (STSG) which can describe the probabilistic mapping (substitution) rules between them. For a given test sentence and the corresponding parsed syntactic tree structure, thus generated STSG can convert the syntactic tree to a prosodic tree statistically. We compare the labeling results with other approaches and show the probabilistic mapping can indeed benefit break index labeling performance.

***Index Terms*—** STSG, break index, TTS

1. INTRODUCTION

Identifying the hierarchical prosodic constituents from text plays an important role in Text-to-Speech (TTS) synthesis. A number of rule-based models and stochastic models are proposed to perform the prosodic analysis. Rule-based approaches were adopted in earlier studies. For example, Joan Bachenko [1] did prosodic phrasing by defining a set of boundary location rules involving syntactic constituency, adjacency to a verb and constituent length. Stochastic models are widely used in more recent studies. Classification and regression tree (CART) using features such as punctuation, part-of-speech (POS) and pitch accent types are adopted to predict break indices in [2], [3] and [4]. Markov model is adopted in [5] and [6]. A more complex hierarchical stochastic model is proposed in [7]. As for break index labeling for Chinese, Min Chu proposed a bottom-up hierarchical approach based on CART using features that could be extracted from text [8]. Jianhua Tao adopted both syntactic features and acoustic features in [9]. Moreover, a more comprehensive comparison was provided in [10].

The break index labeling is generally formulated as a problem of sequential labeling in the conventional approaches. Even though the syntactic tree is used for labeling, only the POS, syntactic phrase types and the word position in phrase, instead of the whole syntactic structure information, are used as features for break labeling. Recently, Mohamed Abou-Zleikha [11] applied tree decomposition to generate pitch contour based on exemplars, where the prosodic-syntactic correlated data and a dynamic unit size model by data-oriented parsing [12] were used. This approach took more syntactic information into account and achieved good results. Thinking in tree structure can help fully exploiting the features of the syntactic structure, such as siblings and ancestors, and thus take a more global view into account when considering inserting breaks. Therefore, we propose to fully take advantage of the syntactic tree structure and predict break indices by studying the probabilistic relationship between the syntactic structure and the prosodic structure. In this paper, we adopt STSG, which is generated statistically to describe the mappings between syntactic and prosodic structures, and use Viterbi search to find the most probable corresponding prosodic structure for the given syntactic structure of the test sentence. Our idea stems from [13], which converts between HPSG (Head-driven Phrase Structure Grammar) and CFG (Context-Free Grammar) based on stochastic STSG.

2. PROSODIC STRUCTURE AND SYNTACTIC PHRASE STRUCTURE

Prosodic structure for the sentence “可喜的是四川正在打破这种意识 (The good news is that Sichuan are breaking this notion)” is depicted as in Figure1(a). Each internal node is labeled with corresponding break indices from 0 to 4 which indicate the duration of breaks between its adjacent child nodes.

Here are the concrete definitions of the break indices [10]. BI0: the non-breaks within a prosodic word; BI1: prosodic word boundary; BI2: minor prosodic phrase boundary; BI3: major prosodic phrase boundary; BI4: prosodic group boundary. Since BI4 is always decided by the end mark of the sentence, we only consider BI0∼BI3.

The syntactic phrase structure for the same sentence is shown in Figure1(b). The internal nodes are labeled with constituent labels indicating the different kinds of phrases they dominate.

From Figure 1 we can see that both are trees, but with different structures. Our study is performed on a corpus, which contains 9,939 sentences with manually labeled break indices and syntactic phrase structures obtained by stanford parser [1] [14]. After excluding the top spans and unary spans, 24,109 out of total 65,043 unique spans in prosodic trees are crossing with the spans of syntactic trees, while 25,598 out of total 55,502 unique spans in syntactic trees are crossing with those of prosodic trees. There are 21,111 spans exactly matched between prosodic trees and syntactic trees. Our approach aims at identifying the mapping relations both crossing and non-crossing between the syntactic structure and the prosodic structure and use these mapping rules to reconstruct the prosodic structure out of a syntactic structure.

3. GENERATING STSG

A STSG rule is defined as a triple <source subtree, target subtree, probability>. The subtree pairs could be extracted as follows. For each pair of tree structures, the nodes that dominate the same span in the pair of trees are identified as the split nodes first and then the subtrees whose external nodes are the split nodes are extracted. As in [13], if the tree pairs include unary productions, the upper-most node of the chain of the unary productions is selected as the split node.

For the latter example in Figure 1, the nodes underlined are split nodes and they segment each tree structure into 13 fragments. Therefore 13 pairs of subtrees are generated and they are listed in Table 1. Subtrees are represented in brackets.

For a STSG rule $< t_1, t_2, p >$, suppose $T\_root(t)$ denote the tree that has the same root label as t's root , then p is defined as

$$p = \frac{Count(< t_1, t_2 >)}{Count(< T\_root(t_1), T\_root(t_2) >)}$$

Therefore, by concatenating the nonterminal leaves of a tree with the roots of other subtrees providing the joint nodes are labeled the same, a larger tree structure could be derived.

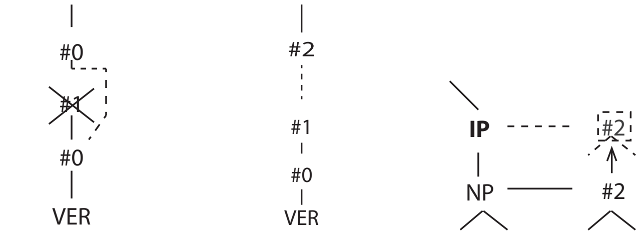
STSG can capture the probabilistic mappings between syntactic and prosodic structures, but there are also other factors which will affect the prosody, for example, the length of the words. Words of only one character tend to form a prosodic word with the character before or after them. Some words such as “的”, “会” also need special attention. In order to take these factors into account, the POS tags are extended with suffix including the length of the word, specifically one or not, and the characters of the word if it belongs to a special group which contains {电, 中, 后, 的, 在, 于, 了, 等, 着, 从, 但是, 也, 还, 被, 不, 目前, 今天, 短波, 简讯, 接着, 就是}.

Table 1. STSG generated for Figure 1.

Syntactic Subtree	Prosodic Subtree	p
(ADJP(JJ))	(0(NOM))	1
(ADVP(AD))	(0(ADV))	1
(DEC)	(AUX)	1
(IP(NP)(VP(ADVP) (VP(VV)(NP))))	(2(1)(1(0)(0))(1))	1
(IP(VP(VA)))	(VER)	1
(NP(NN))	(0(NOM))	0.5
(NP(CP(IP)(DEC)))	(0(VER)(AUX))	0.5
(NP(ADJP)(NP))	(1(0)(0))	0.5
(NP(NR))	(1(0(NOM)))	0.5
(PU)	(3(2(1(0(SYM)))))	1
(ROOT(IP(NP)(VP (VC)(IP))(PU)))	(4(3(2(1(0)(0)))(2))(3))	1
(VC)	(0(VER))	1
(VV)	(0(VER))	1

4. DECODING

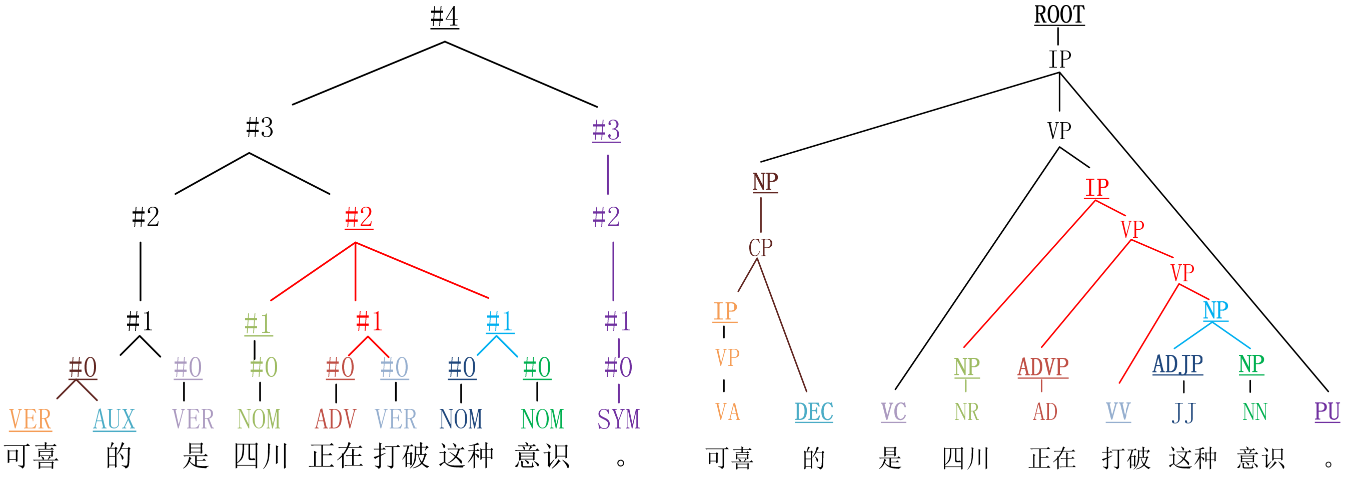
The decoding is to search for the most probable prosodic structure, given the syntactic structure of the test sentence. The probability of a constructed prosodic tree is defined as the product of the applied STSG rules. The syntactic tree is traversed bottom-up and rules that fit both the subtree rooted at the current node of the syntactic tree and the prosodic subtrees already constructed are selected. The algorithm in pseudo code is given in Algorithm 1.



(a)Back-off B (b)Back-off C (c)Back-off E

Fig. 2. Examples of Back-off strategy B, C, and E.

[1] http://nlp.stanford.edu/software/lex-parser.shtml



(a) Prosodic tree structure

(b) Syntactic phrase tree structure

Fig. 1. An example of prosodic tree structure and syntactic phrase tree structure. The nodes underlined are split nodes.

Algorithm 1 Decoding

Input:

The node in the syntactic structure of a test sentence: N

Denotation:

Prt, the prosodic tree structure.

Pht, the syntactic tree structure.

$Rule(Pht, Prt, pr)$, a STSG rule containing the tree pair Pht and Prt, and the probability p of the rule.

$Chart(N) = (Prt_{root \in 0,1,2,3}, p_{root \in 0,1,2,3})$, the chart node of N which contains the corresponding prosodic trees constructed, and each is of the maximum probability among possible prosodic trees with the same root label.

Procedure:

```
for each child node $c$ of $N$ do
    call its decoding procedure to generate Chart(c)
end for
for each $r(Pht, Prt, pr)$ whose $r.Pht$ matches a subtree $st$ rooted at $N$ do
    $p = r.pr$
    for each $st$'s leaf node $i$ do
        Let $l_i$ be the label of the $i^{th}$ leaf node of $r.Prt$.
        Let $c_i$ be the chart node of the $i^{th}$ leaf node of $st$.
        $p = p * c_i.p_{l_i}$
    end for
    Let $l$ be the root label of $r.Prt$
    if $p > Chart(N).p_l$ then
        Let $Chart(N).p_l = p$, $Chart(N).Prt_l = r.Prt$
    end if
end for
if Chart(N) can't be generated then
    Apply back off strategies.
end if
```

5. BACK OFF STRATEGIES

Because the generated STSG rules cannot cover a certain number of source trees, we segment the sentences by punctuation marks, such as “，”, “：”, and “、” to simplify the structure, and add back-off strategies to relax the rules and increase the coverage.

- Strategy A: The POS tags of the syntactic fragment in the rule don't need to match the corresponding POS leaf nodes in the syntactic tree of the test sentence if the POS tags of the prosodic fragment are exactly matched.
- Strategy B: When substituting the leaf node of the prosodic fragment in the rule with the already constructed prosodic root, return match if the leaf node matches the single child of the root, as shown in Figure 2(a).
- Strategy C: When substituting the leaf node of the prosodic fragment in the rule with the already constructed prosodic root, return match if its break index is higher than that of the target root and attach the root to the leaf node, as shown in Figure 2(b).
- Strategy D: Return match if the POS tags of the prosodic fragment's leaf nodes without extended suffix match.
- Strategy E: If no prosodic tree can be constructed for the current syntactic phrase node, copy the constructed prosodic tree of its single child, as shown in Figure 2(c).

6. EXPERIMENTS

The training set contains 8,939 sentences while the test set contains 1,000 sentences. The precision, recall with each or no back off strategy are represented in Table 2.

Table 2. A comparison of the performance when using different back-off strategies.

model	Percent Fail	P&R	BI0	BI1	BI2	BI3
None	12.8	Pre	90.3	72.9	50.4	80.2
		Rec	80.1	83.0	48.3	72.1
A	10.6	Pre	90.2	72.2	50.2	78.3
		Rec	78.3	82.5	48.3	71.0
B	12.4	Pre	90.3	72.8	50.3	79.5
		Rec	79.5	82.9	48.2	72.0
C	7.7	Pre	89.6	71.4	48.9	76.5
		Rec	76.0	81.7	47.7	70.0
D	11.2	Pre	89.4	72.6	49.7	79.3
		Rec	79.6	82.1	47.8	71.9
E	12.5	Pre	90.3	72.8	50.5	80.1
		Rec	79.9	82.9	48.4	72.0
All	4.2	Pre	89.0	70.6	48.1	74.7
		Rec	74.3	80.8	47.4	68.9

Table 2 shows that back-off strategies increase the coverage rate to 95.8% and also lower the performance slightly. By applying all the back-off strategies, we achieved the performance in Tables 3 and 4. The results show that BI2 is more difficult to predict than BI1 and BI3 because BI2 is more ambiguous. BI1 is word boundary and BI3 is generally labeled after a punctuation, while BI2 is the break within a sub-sentence and depends on speaking style, the length of sub-sentence and other various factors.

AvgCost [10] is used as a criterion.

$$AvgCost = \frac{\sum W_i Count(E_i)}{Count(B)}$$

$Count(E_i)$ the number of BI errors equaling i which is defined as the absolute difference between the predicted and the "gold" (given) index. W_i represents the weight for the error E_i. And here we assign $W_1 = 0.5$, $W_2 = 1$,$W_3 = 2$. $Count(B)$ is the count of boundary sites.

From Table 4, we could see that our model performs better than CRF without lexical features, but worse than that of CRF with lexical features. Since no lexical information except a few special words is included in our model, comparing our performance with CRF1, the syntactic tree information captured by STSG indeed improves the performance significantly. Moreover, the difference between the performance of CRF1 and CRF2 shows the importance of lexical information in break index labeling. How to include lexical information into our approach is our future research.

Although the Chinese data sets used in [10] and [8] are not the same as ours, the sizes of the sets are similar (12000 sentences vs 9939 sentences) and the types of break index are also BI0~BI4. Comparing the results, the BI0 in our data is much harder to predict and we obtain a better performance in BI1 and BI3.

Table 3. Confusion matrix using all back-off strategies.

	p0	p1	p2	p3
g0	1298	226	164	60
g1	114	2531	362	124
g2	40	651	818	215
g3	7	169	352	1190

Table 4. A comparison of the performance achieved in predicting breaks with previous results reported in [10] and [8] and CRF. Ours is the test results using all the back off strategies. Ours+bigram uses not only the back off strategies but bigram to predict the failed 4.2% as well. CRF1 and CRF2 are results of CRF++[2] using the same data. CRF1 adopts features including POS and shared phrase ancestor of the adjacent words. CRF2 also uses the lexical features besides those used by CRF1. "-" means not comparable or not available.

model	P&R	BI0	BI1	BI2	BI3	AvgCost
Ours	Pre	89.0	70.6	48.1	74.7	0.19
	Rec	74.3	80.8	47.4	68.9	
Ours+bigram	Pre	87.7	69.5	47.9	74.2	0.18
	Rec	74.6	80.8	45.5	67.8	
CRF1	Pre	74.3	67.3	52.1	74.4	0.22
	Rec	60.3	78.0	46.9	74.8	
CRF2	Pre	89.4	77.4	63.6	81.9	0.14
	Rec	84.1	86.6	56.8	78.6	
Bigram[8]	Pre	81.7	52.5	49.4	60.0	-
	Rec	93.5	45.6	42.4	22.3	
MM[8]	Pre	86.7	56.6	54.6	52.2	-
	Rec	92.9	53.2	49.4	37.8	
C4.5[8]	Pre	90.9	60.3	55.8	50.9	-
	Rec	96.8	58.1	52.8	30.5	
Hierarchical CART[10]	Pre	95.3	65.6	57.4	82.7	-
	Rec	95.7	58.6	65.6	68.1	

Since the parsed trees used are output by a phrase parser automatically and the parsing F measure for Chinese is only around 83%, the error will affect the final results significantly. Moreover, the training data is relatively small comparing with the 40,000 sentences used in [13]. A larger training data set will certainly improve the performance and the coverage.

7. CONCLUSIONS AND FUTURE WORK

We have shown how to take advantage of the structure information between both prosodic and syntactic levels to predict break index. The probabilistic mappings between these two

[2] http://crfpp.googlecode.com/svn/trunk/doc/index.html

structures are captured by STSG and the decoding is to search for the most probable prosodic tree globally. Our results show the syntactic tree structure information and the mapping relations can indeed improve the performance, compared with CRF without using the lexical features.

Since lexical information is very important in predicting break indices, how to incorporate it into the current framework still need to be further investigated. As finitely ambiguous context-free grammars cannot be lexicalized with a tree-substitution grammar [15], lexicalized Synchronous Tree Adjoining Grammars (STAG) might be adopted to take lexical contexts into account. Moreover, since the syntactic trees and the prosodic trees are relatively flat, head-centered binarization of the syntactic structure and binarization of the prosodic tree in the direction that more spans can be shared will help to generate more STSG rules and thus more source trees could be covered during conversion.

8. REFERENCES

[1] Joan Bachenko and Eileen Fitzpatrick, "A computational grammar of discourse-neutral prosodic phrasing in English," *Computational Linguistics*, vol. 16, no. 3, pp. 155–170, Sept. 1990.

[2] Michelle Q. Wang and Julia Hirschberg, "Predicting intonational phrasing from text," in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, USA, June 1991, pp. 285–292, Association for Computational Linguistics.

[3] Julia Hirschberg and Pilar Prieto, "Training intonational phrasing rules automatically for English and Spanish text-to-speech," *Speech Communication*, vol. 18, no. 3, pp. 281–290, 1996.

[4] Sangho Lee and Yung-Hwan Oh, "Tree-based modeling of prosodic phrasing and segmental duration for korean tts systems," *Speech Communication.*, vol. 28, no. 4, pp. 283–300, Aug. 1999.

[5] Nanette M. Veilleux, Mari Ostendorf, Patti J. Price, and Stefanie Shattuck-Hufnagel, "Markov modeling of prosodic phrase structure," in *Proceedings of The 15th International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, New Mexico, USA, apr 1990, ICASSP '90, pp. 777–780.

[6] Paul Taylor and Alan W. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, vol. 12, pp. 99–117, Apr. 1998.

[7] Mari. Ostendorf and Nanette M. Veilleux, "A hierarchical stochastic model for automatic prediction of prosodic boundary location," *Computational Linguistics*, vol. 20, no. 1, pp. 27–54, Mar. 1994.

[8] Min Chu and Yao Qian, "Locating boundaries for prosodic constituents in unrestricted mandarin texts," *Computational Linguistics and Chinese Language Processing*, vol. 6, no. 1, pp. 61–82, Feb. 2001.

[9] Jianhua Tao, "Acoustic and linguistic information based Chinese prosodic boundary labelling," in *Proceedings of The 1st International Symposium on Tonal Aspects of Languages: Emphasis on Tone Languages*, Beijing, China, March 2004, TAL '04.

[10] Yanqiu Shao, Yongzhen Zhao, Jiqing Han, and Ting Liu, "Comparison of approaches for predicting break indices in mandarin speech synthesis," *Journal of Computer Science*, vol. 2, pp. 660–664, 2006.

[11] Mohamed Abou-Zleikha, Peter Cahill, and Julie Carson-Berndsen, "Exemplar-based pitch contour generation using dop for syntactic tree decomposition," in *Proceedings of The 37th International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012, ICASSP '12.

[12] Rens Bod, "Exemplar-based syntax: How to get productivity from examples," *The Linguistic Review*, vol. 23, pp. 291–320, 2006.

[13] Takuya Matsuzaki and Jun'ichi Tsujii, "Comparative parser performance analysis across grammar frameworks through automatic tree conversion using synchronous grammars," in *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, UK, August 2008, COLING '08, pp. 545–552, Coling 2008 Organizing Committee.

[14] Roger Levy and Christopher D. Manning, "Is it harder to parse Chinese, or the Chinese Treebank?," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003, pp. 439–446, Association for Computational Linguistics.

[15] Aravind K. Joshi and Yves Schabes, "Tree-adjoining grammars," in *Handbook of formal languages, vol. 3*, Grzegorz Rozenberg and Arto Salomaa, Eds., pp. 69–123. Springer-Verlag New York, Inc., New York, NY, USA, 1997.