Tree Mapping Template for Prosodic Phrase Boundary Predication

Xun Endong[1], Li Cheng[2]
[1]Beijing Language and Cultural University, No 15 Road Haidian District Beijing 100083
[2]Beijing University of Posts and Telecommunications, 100876
edxun@blcu.edu.cn, alleniversontoplee@peoplemail.com.cn

Abstract

This paper presents a novel method driven by tree mapping template (TMT) which improve the accuracy of prosodic phrase boundary prediction. The TMT is capable of capturing the isomorphic relation between non-terminal nodes in hierarchical prosodic tree and nodes in binary tree approximation, performing pruning at the decoding phase and revising the baseline maximum entropy model with boosting method (AdaBoost). The model is statistical driven because TMTs are extracted automatically from hierarchical prosodic tree and binary tree approximation generated by the Maximum Entropy model. In decoding, TMT is employed to perform pruning, readjusting and local combination. To alleviate data sparse in limited labeled corpus, language model interpolation is made by introducing a large scale unlabeled data. The experiments show that the TMT driven method reduces the decoding complexity for the prosodic phrasing prediction which is crucial for a real-time TTS system, achieving an improvement of 11.5% in terms of F-Measure compared to a conventional maximum entropy model without TMT.

Keywords

Text-to-speech; Binary tree; prosody phrase; Tree Map

1 Introduction

Prosodic phrase boundary prediction (Klatt, 1987; Ostendorf and Vielleux, 1994; Marsi et al., 2003), which aimed to identify the prosodic phrasing and its boundary from an utterance by lexical, syntactic and phonological cues, have been suggested to be one of the most critical part in the real-time production text-to-speech (TTS) system.

In order to achieve natural synthesized speech, the boundary prediction component needs to make critical decisions about the placement of prosodic boundary and possible pause insertion. A variety of prosodic phrases prediction methods based on syntactic structure, rule-based systems, and stochastically learned models have been proposed, making effort to correlate prosodic information with syntactic analyses of text (Ostendorf and Vielleux, 1994; Tao, 2000). However, a key limitation of syntactic-based models is that syntactic phrase dos not directed link to prosodic phrase (Selkirk, 1994), leading to misalignments between the two levels of phrasing.

Recent research on statistics method has lead to the development of prosodic phrase boundary prediction. Sharman and Wright (1996) describe a stochastic parser created on the basis of statistical information, forming manually bracket corpus to predict the phrase boundaries. Wang and Hirschberg (1997) propose a Classification and Regression Tree method, reserving a minimally sized decision tree that is estimated to generalize well to new data. But, this strategy failed to record outliers that take long paths in the tree, which reoccurs in real large corpora. Taylor and Black (1998) represent the training process as a HMM model. Atterer and Klein (2002) treat the prosodic unit as chunks, predicting prosodic phrase under length constraints. Chu and Qian (2001) introduced a hierarchical prosodic tree representation of prosodic phrase, describing local syntactic and length constraint for the structure of prosody in CART model. All these approaches, though different in formalism, make use of local linguistic information to model statistical relations between syntactic phrase and prosodic phrase.

Another class of approaches makes use of rule-based method to generate transducer or automata. Gee and Grosjean (1983) formalizes a number of rules for mapping syntactic structure to a hierarchical representation of phrasing. Hrischberg and Prieto (1996) propose a machine learning method to extract syntactic features from utterance. Zhao et al. (2003) presents a rule constraint method, extracting extended features in both chunk-level and tree-level.

Paying more attention to parsing the hierarchical prosodic structure, Bachenko and Fitzpatrick (1990) employ a binary tree whose terminals are phonological words and whose node labels are indices that mark boundary salience.

In this paper, we propose a prosodic phrase boundary prediction model based on tree mapping template (TMT) which describes the mapping between hierarchical prosodic tree and binary tree approximation. A TMT is capable of capturing the isomorphic relation between non-terminal nodes in hierarchical prosodic tree and nodes in binary tree approximation, integrating Adaboost algorithm (Freund and Schapire, 1996) within the scope of Maximum Entropy model and performing pruning at the decoding phase. The model is statistical driven because TMTs are extracted automatically from hierarchical prosodic tree and binary tree approximation from the Maximum Entropy model. To perform prosodic phrasing prediction, TMT is employed to perform pruning and readjusting.

The remainder of this paper is organized as follows. Section 2 introduced the architecture of the TMT driven method, formally addressing the definition of TMT. Section 3 describes the Maximum Entropy model training with TMT and AdaBoost revision. Section 4 presents the TMT based decoding. Section 5 reports the experimental results. Finally, we made a conclusion and future work discussion in Section 6.

2 Tree Mapping Template

2.1 System Architecture

The prediction of prosodic phrasing boundary has been receiving lasting attention over the years for its important role played in reflecting the naturalness of a concatenated speech synthesis system. The issue could be descried as a pause insertion process (Sharman and Wright, 1996). Given a source string $\tilde{S}$ composed by a word sequence $F_1^{J'}$ and the string context $C_1^{K'}$ (syntactic information, phonological features, and etc.), the optimal pause sequence inserted into the original word sequence that generates the target string $E_1^{I'}$ with the maximum conditional probability:

$$Pause_1^{L'} = \underset{pause}{\arg\max}\left\{Pr\left(E_1^{I'} \mid C_1^{K'}\right)\right\} \tag{1}$$ [1]

Prosodic phrasing boundary prediction in this paper is divided into training phase and decoding phase. A prosodic phrasing boundary labeled corpus is introduced to generate the hierarchical prosodic tree. An adapted language model was trained by using both the labeled and unlabeled corpus. An unlabeled corpus is to alleviate the data sparseness problem raised by limited labeled corpus, at the same time supplying syntactic context (POS tags, word segmentation, and etc.) for the Maximum Entropy model to estimate the distribution of the prosodic phrasing boundaries. A binary tree approximation is made by using a greedy algorithm that searches the pause insertion position in a sentence obtaining the maximum probability in the Adapted LM. A tree-mapping template extracted from the hierarchical prosodic tree and the approximated binary tree, with the divergent branch error rate, was integrating by AdaBoost algorithm into the Maximum Entropy model iteration process. A block diagram of the method used in this paper is presented as Figure 1.

In the decoding phase, the input text is a segmented word sequence POS tags. Then, a binary tree is constructed by the same greedy algorithm in training phase, pruning impossible path of the decoding space. A readjusting and adjoining operation is taken by the tree mapping template to relocate some branches in the binary tree, outputting the text tagged with prosodic phrase boundaries.

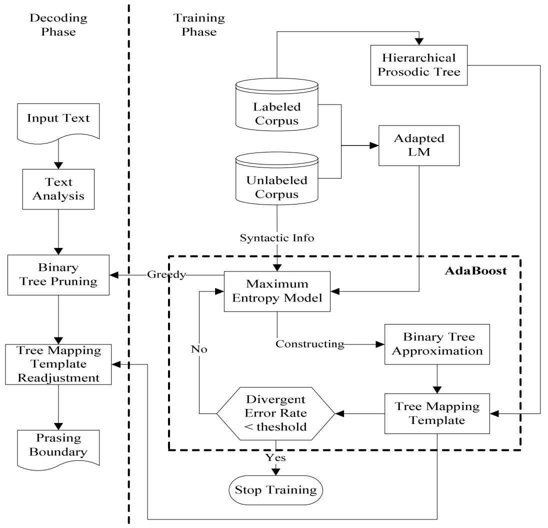


Figure 1, Block Diagram of Training Phrase and Decoding Phrase.

[1] $Pr(\cdot)$ is denoted as the general probability distribution with no specific assumption. Generic symbol $p(\cdot)$ represents for model-based probability distributions.

2.2 Definition of TMT

A Tree Mapping template z is a quadruple $\left\langle \tilde{T}, \tilde{T}_b, \tilde{S}, \tilde{M} \right\rangle$, which describes the mapping (an isomorphic relation) $\tilde{M}$ between a prosodic tree $\tilde{T} = T\left(F_1^{J'}\right)$ [2]and a binary tree approximation $\tilde{T}_b = T_b\left(F_1^{J'}\right)$, an input string (an utterance) denoted as $F_1^{J'}$. An output string $\tilde{S} = E_1^{I'}$ is the sequence of leaf nodes of $T\left(E_1^{I'}\right)$, consisting of both terminals (prosodic words) and non-terminals (prosodic phrase categories, i.e. prosodic phrase and intonational phrase), which is similar to Ladd (1996). A mapping $\tilde{M}$ is defined as a mapping that correlates binary tree node indices with the needed operation for a hierarchical prosodic tree, $i = 0$ presents no operation taken; $i = 1$ stands for adjoining to parent node:

$$\tilde{M} \subseteq \{(j,i) : j = 1,..., J^{'}; i = 0,1\}$$

Figure 2 shows a TMT automatically learned from training data. Note that when demonstrating a TMT graphically, we symbol non-terminals with their categories.

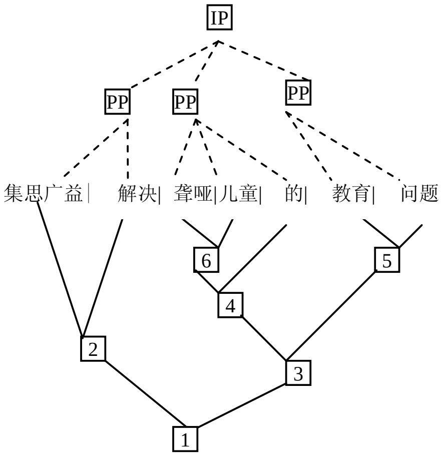


Figure 2. Example of Tree Mapping Template obtained in training.

2.3 Maximum Entropy Model with TMT

In the following, we formally describe how to introduce Tree Mapping templates into probabilistic dependencies to model $Pr\left(e_1^I \mid C_1^K\right)$.

[2] $T(\cdot)$ is to denote the hierarchical prosodic tree. $T_b(\cdot)$ is the binary tree approximation.

In a first step, a hidden variable $T\left(f_1^J\right)$ was introduced that denotes a prosodic tree of the input sentence f_1^J [3]:

$$Pr\left(e_1^I \mid C_1^K\right) = \sum_{T\left(f_1^J\right)} Pr\left(e_1^I, T\left(f_1^J\right) \mid C_1^K\right) \qquad (2)$$

$$= \sum_{T\left(f_1^J\right)} Pr\left(T\left(f_1^J\right) \mid C_1^K\right) Pr\left(e_1^I \mid T\left(f_1^J\right), C_1^K\right) \qquad (3)$$

Next, another hidden variable $T_b\left(f_1^J\right)$ is introduced as an approximation of the hierarchical prosodic tree, decomposing $T\left(f_1^J\right)$ to a binary tree $\tilde{T}_b$ with the same sequence of leaf node. It is assumed that one binary tree $\tilde{T}_b$ produces a sequence of K potential pause point $\tilde{A}_1^K$, inserting into the original string subsequently to generate the output utterance sentence e_1^I with prosodic phrasing tags. In this paper, it is under this assumption that $Pr\left(e_1^I \mid T_b, T\left(f_1^J\right), C_1^J\right) \equiv Pr\left(\tilde{A}_1^K \mid T_b, T\left(f_1^J\right)\right)$ because e_1^I is actually generated by the derivation of pause point $\tilde{A}_1^K$. Note that we omit an explicit dependence on the approximation T_b to avoid notational overhead.

$$Pr\left(e_1^I \mid T\left(f_1^J\right), C_1^J\right) = \sum_{T_b} Pr\left(e_1^I, T_b \mid T\left(f_1^J\right), C_1^J\right) \qquad (4)$$

$$= \sum_{T_b} Pr\left(T_b \mid T\left(f_1^J\right), C_1^J\right) Pr\left(e_1^I \mid T_b, T\left(f_1^J\right), C_1^J\right) \qquad (5)$$

$$= \sum_{T_b} Pr\left(T_b \mid T\left(f_1^J\right), C_1^J\right) Pr\left(\tilde{A}_1^K \mid \tilde{T}_b, T\left(f_1^J\right)\right) \qquad (6)$$

$$= \sum_{T_b} Pr\left(T_b \mid T\left(f_1^J\right), C_1^J\right) \prod_{k=1}^{K} Pr\left(\tilde{A}_k \mid \tilde{T}_b, T\left(f_1^J\right)\right) \qquad (7)$$

To further decompose $Pr\left(\tilde{A} \mid \tilde{T}_b, T\left(f_1^J\right)\right)$, the Tree Mapping template, denoted by the variable z, is introduced as a hidden variable.

$$Pr\left(\tilde{A} \mid \tilde{T}_b, T\left(f_1^J\right)\right) = \sum_{z} Pr\left(\tilde{A}, z \mid \tilde{T}_b, T\left(f_1^J\right)\right) \qquad (8)$$

$$= \sum_{z} Pr\left(z \mid \tilde{T}_b, T\left(f_1^J\right)\right) Pr\left(\tilde{A} \mid z, \tilde{T}_b, T\left(f_1^J\right)\right) \qquad (9)$$

A further assumption could be made $Pr\left(\tilde{A} \mid z, \tilde{T}_b, T\left(f_1^J\right)\right) \equiv Pr\left(\tilde{A} \mid z, \tilde{T}_b\right)$ because z and $\tilde{T}_b$ contains sufficient information to deduce $T\left(f_1^J\right)$. $Pr\left(T\left(f_1^J\right) \mid C_1^J\right)$ is constant

[3] The mathematical notation we use in this paper: an input string $f_1^J = f_1, \ldots, f_j, \ldots, f_J$ with length J is to be inserted with pause point to an output string $e_1^I = e_1, \ldots, e_i, \ldots, e_I$ with length I, C_1^K is the context and $Pause_1^L$ is the pause point and so forth.

because one input string corresponds to only one hierarchical prosodic tree. Therefore, the TMT-guided prediction model can be decomposed into 3 sub-models:

1. Approximation model: $Pr\left(T_b \mid T\left(f_1^J\right), C_1^J\right)$
2. TMT Training model: $Pr\left(z \mid \tilde{T}_b, T\left(f_1^J\right)\right)$
3. TMT Decoding model: $Pr\left(\tilde{A} \mid z, \tilde{T}_b\right)$

Following Berger et al. (1996), we base our model on Maximum Entropy framework. Hence, all knowledge sources are described as feature functions that include the given source string f_1^J, the target string e_1^I, and syntactical and phonological context. The hidden variable $T\left(f_1^J\right)$ is omitted because we usually make use of only single prosodic hierarchical tree. As we assume that the binary tree approximation for the prosodic tree have the same probability, the hidden variable T_b is also omitted. As a result, the model we actually adopt for experiments is limited because the tagging, approximation, and TMT deployment sub-models are simplified.

$$Pr\left(e_1^I, z_1^J \mid C_1^K\right) = \frac{\exp\left[\sum_{i=1}^{n} \lambda_i h_i\left(e_1^I, C_1^K, z_1^J\right)\right]}{\sum_{e_1'^I, z_1'^J} \exp\left[\sum_{i=1}^{n} \lambda_i h_i\left(e_1'^I, C_1^K, z_1'^J\right)\right]}$$

The following seven feature functions using both heuristics and contextual information similar as in Chu and Qian (2001). To simplify the notation, we omit the dependence on the hidden variables of the model. First, an adapted language model was introduced:

A human labeled corpus with prosodic boundary tags and a large scale corpus from general domain were introduced. Thus, a language model with the boundary tags $P_L(W)$ (ULM) and a general large scale language model $P_U(W)$ (ULM) were built respectively.

$$P(W) = \lambda \times P_U(W) + (1-\lambda) \times P_L(W)$$

Where λ is a linear interpolation weight, here the adapted LM feature function was defined:

$$h_1\left(e_1^I, C_1^K\right) = \log \prod_{i=1}^{I} P\left(w_i \mid w_{i-1}, w_{i-2}\right)$$

Word frequency of the current word:

$$h_2\left(e_1^I, C_1^K\right) = count\left(w_{curr}\right)$$

Word length of previous constituent:

$$h_3\left(e_1^I, C_1^K\right) = L$$

Intonational Category of Pinyin (Bopomofo), i.e. 3 represents for antichlor:

$$h_4\left(e_1^I, C_1^K\right) = \{Y \mid Y = 1, 2, 3, 4\}$$

POS tagging trigram:

$$h_5\left(e_1^I, C_1^K\right) = \log \prod_{i=1}^{I} P\left(pos_i \mid pos_{i-1}, pos_{i-2}\right)$$

Number of TMT template used:

$$h_6\left(e_1^I, C_1^K\right) = K,$$

Branch divergent penalty function:

$$h_7\left(e_1^I, C_1^K\right) = \sum_{k=1}^{N} Dive\left(T_k, \tilde{T}_{bk}\right)$$

$$Dive\left(T_k, \tilde{T}_{bk}\right) = 1 - \frac{2\left|T_k \wedge \tilde{T}_{bk}\right|}{\left|T_k\right| + \left|\tilde{T}_{bk}\right|}$$

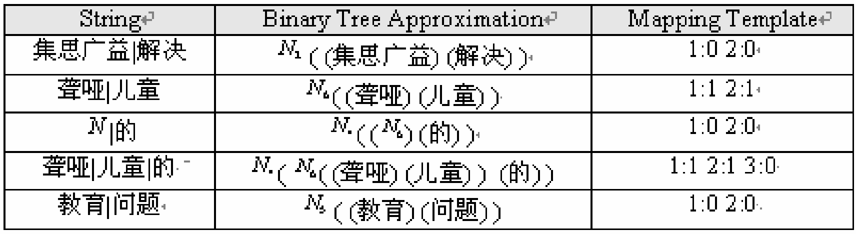
where N is the tree pair number, $\left|T_k \wedge \tilde{T}_{bk}\right|$ represents the number of branches that the hierarchical prosodic tree and binary tree approximation shared with the same direct parent node. It could be obtained by using a post-order transversal in both trees. Hence, the Divergent Branch Penalty Rate could be obtained in training corpus.

The Branch divergent penalty feature function is introduced because prosodic boundary across branches is computational costly to predict in the decoding stage.

3 Training

To extract tree mapping templates from a prosodic hierarchical tree and a binary tree, a set of constraint should be detailed addressed:

1. $\forall (i, j) \in \tilde{M}, i_1 \le i \le i_2, j_1 \le j \le j_2$



String	Binary Tree Approximation	Mapping Template
集思广益|解决	N_1（(集思广益) (解决)）	1:0 2:0
聋哑|儿童	N_4（(聋哑) (儿童)）	1:1 2:1
N|的	N.（(N_4) (的)）	1:0 2:0
聋哑|儿童|的	N.（ N_4((聋哑) (儿童)） (的)）	1:1 2:1 3:0
教育|问题	N_5（(教育)(问题)）	1:0 2:0

Table 1: Example of TMTs extracted from in Figure 1 with h = 2

2 $Leaf\left(T\left(e_i^j\right)\right) \wedge Leaf\left(T_b\left(e_i^j\right)\right) \ne \varnothing$

3. Leaf node or non-terminal node representing prosodic word(s) should share the same direct parent node.

4. The height of $T_b\left(e_i^j\right)$ is no greater than h.

Constraint 4 is introduced for the efficiency purposes. Under constraint 3, a kind of mapping that the pause points are in different branches sharing no direct parent node will not be extracted. Considering that some useful mapping templates could be lost, the divergent branch penalty feature function h_6 is introduced to punish those binary trees that falsely split the entire prosodic boundary into separate part in the upper level of the tree.

A binary tree was initially constructed by using the following greedy algorithm:

```
1: Input: a segmented word sequence (W), and start and end offset (L, H), boundary array (R)
2: Return: the position of the prosodic boundary
3: Proc ContructBinaryTree(W, L, H, R)
4:     pos = 0
```

```
    if H - L < 1 then
            add pos to R; return pos;
    end if
    for each pause point pos from L to H-1 do
        finding the pause insertion position pos that obtain the maximum probability of
          the sentence;
    end for
    add pos to R;
  return ContructBinaryTree(W, L, L+pos-1, R);
  return ContructBinaryTree(W, L+pos+1, H, R);
  end Proc
  Output: boundary array R
```

Figure 3. Greedy algorithm for constructing binary tree.

The algorithm for constructing the binary tree is a greedy method in nature in that a local optimal pause point may be found instead of a global optimal pause point by this algorithm. Hence, We risk a situation where the prosodic phrase could be segmented at the upper level of the binary tree, violating constraint 3 and making readjustment and local adjoining more difficult in the decoding phase. For example, supposing leaves of node 2 and node 6 in Figure 1 were within the same prosodic phrase, TMT containing node 2 and node 6 will not be extracted under current constraints, leading to error in decoding phase. A problem on the divergent branch split across prosodic boundaries is fully considered because it is more error-prone at the decoding phase. An Adaboost algorithm (Freund and Schapire, 1996) applied with the branch divergent penalty feature function is therefore developed to reduce the occasions which violate constraint 3. It could be described as Figure 4:

```
Input: A training set includes $m$ - pairs hierarchical prosodic tree $T(S)$ and binary
  tree $T_b(S)$. $d_i^t$ is the distribution weight of $i$-th tree pair at iteration stage $t$. $h(t)$ is
  the distribution probability estimated in the Maximum Entropy model, and its weight
  is $\alpha(t)$. $L$ is the loop count.
Proc AdaBoostForMaximumEntropy
Initialize $d_i^t$ =1/m,
for $t$ = 1 to $L$ do
  for each tree pair weight do
          $p_i(t) = d_i^t / \sum_t d_i^t, t = 1, 2, \ldots, m$
  end for
  Calculate the divergent branch error rate $\varepsilon_t$ (DBER) from tree mapping template.
    $\varepsilon_t = \sum_{i=1}^{m} p_t(i)\alpha(i)/m$, where $\alpha(i)$ is calculated as defined in feature function $h_7$.
  if $\varepsilon_t$ >= 1/2 or $\varepsilon_t$ = = 0 then,
    set $t = L$ -1 and exit
  end if
```

```
set $\beta(t) = \log((1-\varepsilon_t)/\varepsilon_t)$
    for each $t$ do
        $d_i^{t+1} = d_i^t \exp\{-\beta(t)h(t)\}/Z_t$
        $Z_t = \sum_{i=1}^{m} d_i^t \exp\{-\beta(t)h(t)\}$, updating weight fortemplate of tree pair $< T(S), T_b(S) >$.
    end for
end for
end Proc
Output: The revised distribution trained in Maximum Entropy model $h_L = \sum_{t=1}^{L} \beta(t)h(t)$.
```

Figure 4. Adaboost algorithm based on Divergent Branch Error Rate Learning for MaxEnt model revision.

4 Decoding

The decoding problem was approached as a construction of a binary tree described in section 2.

Given an input text $W = w_1 w_2 ... w_n$, a $n-1$ potential pause point could be found theoretically. A candidate prosodic boundary could be the pause point position that obtains the maximum sentence probability after the pause point insertion:

$$Pause_1^{I'} = \tilde{A}_1^K = \arg\max_{pause} \left\{ Pr\left(e_1^{I'} = W' \mid C_1^{K'}\right) \right\}$$

$$W' = w_1 w_2 ... w_{i-1} \Delta w_i ... w_n$$

The sentence will be split to two parts by the candidate prosodic boundary to form a binary tree.

We could use the same greedy algorithm in training phase to segment the word sequence recursively into a binary tree. The TMT template in the training phase could be used for readjusting and local adjoining.

A TMT z is *deployable* if and only if $T(z)$ covers parts of the nodes of the binary tree T_b. Given a T_b, if T_b was found in TMT z, then T_b is *deployed* to $T(z)$. The TMT could be used for pruning for the input sequence which does not correspond to a binary tree T_b. Figure 5 shows the procedure of decoding with TMT. First, the input sentence is split to a binary tree. A binary tree approximation was made on the input segmented sequences to make pruning of the potential boundaries, discarding impossible path in decoding space. Next, TMTs extracted from hierarchical prosodic tree was deployed to conduct the readjustment and local adjoining from binary tree to hierarchical prosodic tree. Finally, non-terminal nodes are combined serially to generate the output string with prosodic boundary labels.

In Comparison, a viterbi beam search algorithm is developed for the Maxent model without TMT.

5 Experiments

In this section, we report on experiments with Tree Mapping Template driven prosodic phrase boundary prediction. The adapted LM described in section 3.1 was an interpolation of large scale unlabeled data (ULM) with 473, 8103 sentences and limited boundary labeled data (LLM) with 3,000 sentences. The labeled data was composed by a selection of *People's daily* in 1998, containing 9.17 words and 3.68 pause point in each sentence. All sentences were segmented into word sequence with Part-of-Speech annotation. Prosodic phrase boundary in LLM corpora were manually annotated phonological phrasing categories (IP, PP). The hierarchical prosodic tree comes from the labeled data, while the binary tree in decoding phase was constructed from the probability distribution estimated of maximum entropy model.

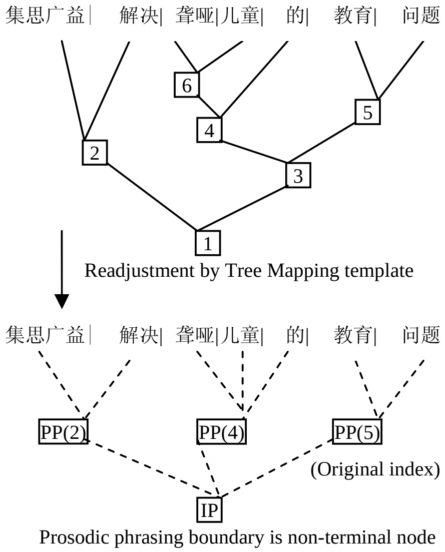

集思广益| 解决| 聋哑|儿童| 的| 教育| 问题
6
4
5
2
3
1
Readjustment by Tree Mapping template
集思广益| 解决| 聋哑|儿童| 的| 教育| 问题
PP(2)
PP(4)
PP(5)
(Original index)
IP
Prosodic phrasing boundary is non-terminal node

Figure 5: Decoding with TMT pruning.

One in ten of the LLM corpus was left out for testing corpus, and the rest were development corpus. For the language model, we used SRI Language Modeling Toolkit to train a trigram model for ULM and a bigram model for LLM with modified Kneser-Ney smoothing (Chen and Goodman, 1998) on the corpora.

We evaluated the prosodic phrase boundary prediction performance by using precision, recall, F-Measure and Divergent Branch Error Rate (DBER) metrics described in section 3.

Method		Precision	Recall	F-Measure	DBER
Labeled LM (Only)	MaxEnt with beam search	0. 5627	0.5094	0.5347	—
	MaxEnt with TMT	0.6103	0.5522	0.5798	0.3022
	MaxEnt (AdaBoost revised) with TMT	0.6180	0.5611	0.5882	0.2884
Adapted LM	MaxEnt with beam search	0.6213	0.6929	0.6551	—
	MaxEnt with TMT	0.7452	0.7184	0.7316	0.3739
	MaxEnt (AdaBoost revised) with TMT	0.7986	0.7428	0.7697	0.2910

Table 2: Comparison of baseline system and TMT-based approach with different method settings

5.1 TMT in MaxEnt Model

The baseline system used for comparison is a prediction by viterb beam search decoding algorithm. A lattice was built upon the probability distribution estimated from maximum entropy model. The beam width is set to 5.For the conventional maximum entropy model, six features were used. In TMT-based decoding, binary tree approximation was made to cut impossible path in decoding space. TMT extraction described in section 3 was performed with h=3, obtaining 4,613 TMTs in LLM and 5,780 in ULM. A leave-one-out method was adopted and 300 sentences in the LLM were left for testing. A cross-validation test was taken for evaluation. We find that the TMT method outperforms the baseline system with an improvement in terms of F-Measure of 4.51% in labeled data and 7.65% in adapted LM as Table 2 shown.

In decoding, we observed that 423 TMTs have been deployed to readjustment of binary tree in LLM and 519 TMTs in ULM, 1.4 times for each sentence in average in LLM and 1.7 times in ULM.

5.2 AdaBoost in MaxEnt Model

The AdaBoost Algorithm is proposed under constraints in section 3 that divergent branches were egregious for both training and decoding. Like Bachenko and Fitzpatrick (1990), we focus on prosodic phrases that are falsely segmented by the binary tree in the upper level.

In the experiments, the maximum iteration time was limited to 50 for efficiency purpose. The DBER has fallen from 0.3022 to 0.2884 in labeled data and from 0.3739 to 0.2910 in adapted LM, reducing DBER to a satisfying level as expected. Binary tree in this paper was applied in both training (revising MaxEnt model) and decoding, whilst Bachenko and Fitzpatrick (1990) propose the binary as a decoding strategy only.

5.3 Unlabeled Data in MaxEnt Model

It is interesting to discuss the relation of the training sentence and its corresponding binary tree in the adapted LM because it shows the pruning performance of the binary tree approximation in large corpus. Coverage recall was defined as the percentage of sentence in training data could maps to a binary tree. The coverage recall of binary tree in the training data varies with linear interpolation weight λ in the Adapted language model.

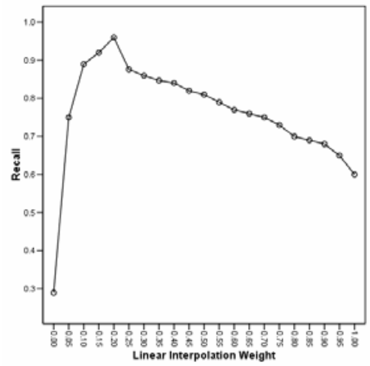


Figure 6. Coverage recall of binary tree coverage under different interpolation weight.

In the experiments, an optimal λ corresponding to a coverage recall of 96% was obtained at 0.175, compensating for two types of data. The benefit of the unlabeled large scale corpus lies in covering more linguistic phenomena that are helpful in the identification of named entities, dates and numbers, generalizing the limited data to more general domain data. The coverage recall of binary tree in the training set is shown in Figure 6.

6 Conclusion

In this paper, a Tree Mapping Template (TMT) is introduced, which can be automatically learned from prosodic hierarchical tree and a binary tree approximation to improve the performance of prosodic phrasing boundary prediction. Compared to conventional statistical method, a binary tree approximation is introduced for the purpose of real-time requirement and decoding efficiency consideration. An AdaBoost algorithm is developed to revise the Maximum Entropy model iteratively by divergent branch penalty feature function. A revised Maximum Entropy model is applied in the construction of binary tree in decoding phase with local branch adjoining to generate the final prosodic phrasing. The experiments report a satisfying result of 11.5% improvement in terms of F-Measure.

It should be emphasized that the constrictions imposed on TMT extraction limit its expressive capability of the prosodic phrasing maps between trees. Preliminary experiments reveal that removal of these constrictions could improve the F-Measure further but at an expense of high time consumption. A balance of time expense and system accuracy should be further thought through in the future work.

7 References

Michaela Atterer and Ewan Klein. 2002. *Integrating linguistic and performance2based constraint s for assigning phrase breaks*. In Proceedings of 17th COLING, Taipei.

A.L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. *A maximum entropy approach to natural language processing*. Computational Linguistics, 22(1):39–72, March.

J. Bachenko and E. Fitzpatrick. 1990 *A Computational Grammar of Discourse-Neutral Prosodic Phrasing in English.* Computational Linguistics, vol 16, pp 155-170.

N.Campbell. *Automatic detection of prosodic boundaries in speech*, Speech Communication, vol 13, pp 343-354, North-Holland, 1993

M. Chu, Y. Qian, *Locating Boundaries for Prosodic Constituent s in Unrestricted Mandarin Texts*, Computational Linguistics and Chinese Language Processing , Vol. 6 , No. 1 , February 2001 , pp. 1222.

Yoav Freund and Robert E. Schapire. 1996. *Experiments with a New Boosting Algorithm.* In Proc. Of the 13th International Conference on Machine Learning (ICML-1996), pages 148-156.

Stanley F. Chen and Joshua Goodman. 1998. *Am empirical study of smoothing techniques for language modeling*. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

J. Hirschberg. 1993. *Pitch accent in context: Predicting intonational prominence from text*. Artificial Intelligence, 63:305–340.

J. P Gee. and F. Grosjean 1983 *Performance Structures: A Psycholinguistic and Linguistic Appraisal*. Cognitive Psychology 15:411-458.

J. Hrischberg and P. Prieto. 1996. *Training Intonational phrasing rules automatically for English and Spanish text-to speech.* Speech Communication.Vo1.18.1996.pp.28 1-290.

P. Koehn, S. Abney, J. Hirschberg, and M. Collins. 2000. *Improving intonational phrasing with syntactic information*. In ICASSP, pages 1289–1290.

D. H. Klatt. 1987. *Review of Text-to-Speech Conversion for English*. Journal of the Acoustic Society of America 82: 737-793.

D. R. Ladd. 1996. *Intonational phonology*. Cambridge University Press.

Erwin Marsi; Martin Reynaert; Antal van den Bosch; Walter Daelemans; Véronique Hoste 2003. *Learning to Predict Pitch Accents and Prosodic Boundaries in Dutch*. In Proceedings of the ACL 2003.

M.Ostendorf and N.Vielleux. *A hierarchical stochastic model for automatic prediction of prosodic boundary location*. Computational Linguistics, vol 20, pp 27-54, 1994

E. O Selkirk. 1984 *Phonology and Syntax: The Relation between Sound and Structure.* MIT Press, Cambridge, MA.

Richard A.Sharman and Jery H. Wright. 1996. *A Fast Stochastic Parser For Determining Phrase Boundaries For Text-To-Speech Synthesis.* In Proceeding of 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP-96).

Paul Taylor and Alan W Black. (1998) *Assigning phrase breaks from part of-speech sequences*. Computer Speech and Language v12.

J.H. Tao. 2000. *Rhythm of Spoken Chinese – Linguistic and Paralinguistic Evidences.* In Proc. of ICSLP 2000, Beijing, pp. 357-360.

Sheng Zhao, Jianhua, Tao, DanLing Jiang . 2003. *Chinese prosodic phrasing with extended features*. In Proceeding of 2003 IEEE International conference on Acoustics, Speech, and Signal Processing, (ICASSP 2003)

Michelle Wang and Julia Hirschberg. 1992. *Automatic classification of intonational phrase boundaries.* Computer Speech and Language 6: 175-196.