ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Expand CRF to Model Long Distance Dependencies in Prosodic Break Prediction

Jian Luan[1], *Bolei He**, *Hairong Xia*[1], *Linfang Wang*[1], *Braga Daniela*[2], *Sheng Zhao*[1]

[1]Speech Team OSD, Microsoft (China) Corp., Beijing, China
[2]Information Platform & Experiences Group, Microsoft Corp., Redmond, WA, USA
jianluan@microsoft.com

Abstract

Intonation phrase length distribution is important information for prosodic break prediction. However, existing CRF frameworks cannot make full use of it. An expanded CRF is proposed in this paper to tackle this problem. Its lattice carries the location of previous intonation phrase (L3) break, and consequently makes it possible to support various dynamic features, such as the number of syllables from the previous L3 break and the POS of word after the previous L3 break. Remarkable improvements are obtained with the expanded CRF for L3 break prediction task. It is also promising to benefit other tasks containing long distance dependencies.

Index Terms: CRF, intonation phrase, prosodic break prediction, speech prosody

10.21437/Interspeech.2012-655

1. Introduction

Prosody is the phonological structure of spoken language and plays an important role in expressing the meaning and emotion. Generally, it is considered to include rhythm, stress and intonation. To represent it in TTS, many sets of prosody feature symbols have been designed. Most of them agree on a hierarchical structure, which control the rhythm directly and locate various pitch/tone accents. A common prosody structure has four layers: word, intermediate phrase, intonation phrase and sentence. They are segmented by corresponding breaks, written L1, L2, L3 and L4 respectively hereafter for the sake of convenience.

Many studies were reported on prosodic break prediction. Recently, Conditional Random Fields (CRF) [1] is introduced into this field and obtains good results [2][3]. In comparison to the conventional Classification and Regression Tree (CART) and Maximum Entropy (ME), CRF introduces Markov chain to take the relationship between neighboring prosodic breaks into account and outputs the optimal break sequence as a whole rather than making decisions independently. Yet, the conventional (linear-chain) CRF is still has some shortcoming for this task. In particular, the distribution of intonation phrase has been proved to be very helpful in L3 break prediction (J. Li et al. 2005) [4]. Since the two L3 breaks at the boundary of an intonation phrase are usually too far away from each other, their dependencies generally become very weak in the linear-chain CRF with only bigram feature.

In other fields, efforts have been made on CRF to cope with similar problems. Semi-Markov CRF (Sarawagi and Cohen, 2004) [5] was proposed for Name Entity Recognition (NER) task. It groups the continuous units with the same labels and replaces the product of all transition probability inside one group with a function of group length and label type. In this way, the length information is explicitly modeled, that seems to match our intention. However, it should be noted that Semi-Markov CRF is not flexible enough. In the case of L3 prediction, the intonation phrase length counted in syllable is as important as or more important than that counted in word. Semi-Markov CRF cannot support the former feature but the latter one. Skip-chain CRF (Sutton and McCallum, 2004) [6] admits to skip some units in the sequence and focus only on key units that strongly influence each other. It can treat two distant labels as neighbors in the sequence to build the long distance dependencies explicitly with simple bigram. However, the original length between two labels in the skipped chain is usually not applied and the information carried by skipped units is ignored as the same time. Factorial CRF (F-CRF) (Sutton et al., 2007) [7] adds another set of symbols as medium layer to model the dependency of one output label on some remote input features. It is not suitable to our task, because it aims not at the relationship between two output labels, neither their label types nor their distance.

In this paper, we expand the linear-chain CRF to model the long distance dependencies on output labels. In the expanded CRF framework, one output label type is regarded as key and all other types are supposed to have long distance dependencies on the previous key label. Various features can be designed to model such dependencies. E.g. sum of a numeric feature of each unit in the range from previous key label to current one, a feature around the previous key label and all their combinations. Since these kinds of features rely on the location of the previous key label and the location is not determined yet in Viterbi searching, they are called dynamic features hereafter. Several dynamic features are tested in our experiment and achieve obvious improvement on L3 break prediction. On the other hand, the computation cost of the expanded CRF is on par with Semi-Markov CRF, proportional to the maximal length L and much cheaper than L-order CRF. Yet expanded CRF provide more choices than Semi-Markov CRF in modeling the long distance dependencies with various dynamic features.

The rest of the paper is organized as follows: in section 2, the theory of linear-chain CRF is reviewed. Section 3 then introduces the expanded CRF. Experiment results are shown in Section 4, followed by the description of speech corpus and feature template. Finally, conclusions are made in section 5.

2. Linear-chain CRF

Conditional Random Fields (CRF) are probabilistic models for computing the probability of a possible output label sequence $Y=(y_1, y_2, \ldots, y_T)$ given the input feature sequence $X=(x_1, x_2, \ldots,$

* Worked as an intern in Speech Team, Microsoft Asia R&D Group.

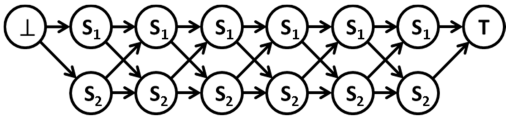

Figure 1 Linear-chain CRF

x_T). In comparison to the classical CART and ME, CRF models the relationship not only between one output label and its context input features but also between one output label and its neighboring labels. Thus, it shows better performance in many tasks. In essence, CRF can be understood as the ME model linked by Markov chain.

2.1. Basic theory

A general form of CRF is structured as a bigram linear-chain, as shown in Figure 1. Each column stands for a unit in the sequence while each node in a column represents one possible label for current unit. There are only two label types, s_1 and s_2, are drawn in the lattice. ⊥ and T are dumb start and end labels respectively.

Each node carries an emission probability, the joint probability between *t*-th label y_t and input feature sequence X at the location *t*:

$$\mathrm{p}(y_t, X, t) = \exp\left(\sum_{m=1}^{M} \lambda_m f_m(y_t, X, t)\right) \quad (1)$$

where, M is the number of feature functions f. λ_m is the weight for *m*-th unigram feature function f_m. And each arc carries a transition probability:

$$\mathrm{p}(y_t, y_{t-1}) = \exp\left(\sum_{n=1}^{N} \psi_n g_n(y_t, y_{t-1})\right) \quad (2)$$

where, N is the number of feature functions g. ψ_n is the weight for *n*-th bigram feature function g_n. Accordingly, the conditional probability of a label sequence Y given input feature sequence X is:

$$\mathrm{p}(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \mathrm{p}(y_t, X, t)\, \mathrm{p}(y_t, y_{t-1}) \quad (3)$$

where, T is the sequence length. The normalization factor $\mathrm{Z}(X)$ for input feature sequence X is the summation of $\mathrm{p}(Y|X)$ over all possible label sequence $\mathfrak{Y}$:

$$\mathrm{Z}(X) = \sum_{Y \in \mathfrak{Y}} \mathrm{p}(Y|X) \quad (4)$$

2.2. Training

The purpose of training is to obtain the unigram feature weights $\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_N)$ and bigram feature weights $\Psi = (\psi_1, \psi_2, \cdots, \psi_M)$ maximizing the summation of $\mathrm{p}(Y|X)$ on all the training data. The forward-backward algorithm is used here. In Figure 1, a node locating at *i*-th column and *j*-th row of the lattice denotes that *i*-th label in the sequence y_i is the *j*-th type s_j in the label collection S. Then, its forward probability $\alpha_{i,j}$ and backward probability $\beta_{i,j}$ are:

$$\alpha_{i,j} = \sum_{k=1}^{J} \alpha_{i-1,k} \mathrm{p}(s_j, s_k) \mathrm{p}(s_j, X, i) \quad (5)$$

$$\beta_{i,j} = \sum_{k=1}^{J} \beta_{i+1,k} \mathrm{p}(s_k, s_j)\, \mathrm{p}(s_j, X, i) \quad (6)$$

where, J is the size of label set S, $\mathrm{p}(s_j, X, i)$ is the emission probability of current node, $\mathrm{p}(s_j, s_k)$ and $\mathrm{p}(s_k, s_j)$ are transition probabilities from previous column to current one and from the current to the next respectively.

With the forward and backward probability, the gradient for each weight can be easily calculated by:

$$\lambda_n = \left(\tilde{E}(f_n) - E(f_n)\right) \cdot \sigma^2 \quad (7)$$

$$\psi_m = \left(\tilde{E}(g_m) - E(g_m)\right) \cdot \sigma^2 \quad (8)$$

where, $\tilde{E}(f_n)$ and $\tilde{E}(\psi_m)$ are expectations of feature function f_n and g_m in the training data. They are constants in the training. $E(f_n)$ and $E(g_m)$ are expectations of f_n and g_m by the latest CRF model. Each iteration need to update them based on the forward and backward probability calculation. More details about the training may refer to [1][8].

2.3. Inference

Inference is to find the most likely label sequence Y for given feature sequence X. As a classical dynamic programming algorithm, Viterbi searching is applied here. Its main difference from the forward-backward algorithm is that Viterbi aims only at the optimal label sequence rather than the sum of all possible sequences.

3. Expanded CRF

In some tasks, the dependency between two output labels far away from each other is important. However, it cannot be explicitly modeled with general bigram transition feature. Supposing the longest distance of such dependency is L, then L-order CRF has to applied and the computation complexity grows exponentially as L increases. An expanded framework is proposed below for CRF to cope with this problem.

3.1. Theory

To simplify the problem, only one label type, written s_1, is supposed to have long distance impact on other labels following it. It is reasonable in most cases. Even for a complex task, it can be divided into several simple tasks to ensure each task contains only one key label type.

A new component for modeling the long-distance dependencies is added in the conditional probability $\mathrm{p}(Y|X)$ as:

$$\mathrm{p}(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \mathrm{p}(y_t, X, t)\, \mathrm{p}(y_t, y_{t-1}) p(y_t, X, t, d) \quad (9)$$

with

$$\mathrm{p}(y_t, X, t, d) = \exp\left(\sum_{o=1}^{O} \varphi_o h_o(y_t, X, t, d)\right) \quad (10)$$

where, d is the distance from the previous key label s_1 to current (t-th) label, O is the number of dynamic feature functions h and φ_o is the weight for o-th dynamic feature function h_o.

3.2. Training and Inference

In order to exhibit the information d, conventional rectangle lattice (see Figure 1) is extended into a triangle lattice (as shown in Figure 2). In each column, there is still unique node for s_1. However, other label types may possess multiple nodes at this time. For the convenience of description, only two label types are drawn in Figure 2. One is key type s_1 and the other is non-key type s_2. The multiple nodes representing s_2 in one column are to carry different distance from the previous s_1. Thus, in the new lattice, a node for non-key types needs three indices i, j and d to locate it, i.e. in i-th column, belongs to s_j and d columns away from previous s_1.

Then, the forward and backward probabilities for key type s_1 are:

$$\alpha_{i,1} = \mathrm{p}(s_1, X, i) \sum_{k=2}^{J} \sum_{d=1}^{i} \alpha_{i-1,k,d}\, \mathrm{p}(s_1, s_k)\mathrm{p}(s_1, X, i, d+1) + \mathrm{p}(s_1, X, i)\alpha_{i-1,1}\mathrm{p}(s_1, s_1)\mathrm{p}(s_1, X, i, 1) \quad (11)$$

$$\beta_{i,1} = \mathrm{p}(s_1, X, i) \sum_{k=2}^{J} \beta_{i+1,k,1}\mathrm{p}(s_k, s_1)\mathrm{p}(s_k, X, i+1,1) + \mathrm{p}(s_1, X, i)\, \beta_{i+1,1}\mathrm{p}(s_1, s_1)\mathrm{p}(s_1, X, i+1,1) \quad (12)$$

The forward and backward probabilities for non-key type s_j (j>1) are:

If d=1

$$\alpha_{i,j,d} = \mathrm{p}(s_j, X, i)\alpha_{i-1,1}\mathrm{p}(s_j, s_1)\mathrm{p}(s_j, X, i, 1) \quad (13)$$

Else

$$\alpha_{i,j,d} = \mathrm{p}(s_j, X, i) \sum_{k=2}^{J} \alpha_{i-1,k,d-1}\mathrm{p}(s_j, s_k)\mathrm{p}(s_j, X, i, d) \quad (14)$$

$$\beta_{i,j,d} = \mathrm{p}(s_j, X, i)\, \beta_{i+1,1}\mathrm{p}(s_1, s_j)\mathrm{p}(s_1, X, i+1, d+1) + \mathrm{p}(s_j, X, i) \sum_{k=2}^{J} \beta_{i+1,k,d+1}\mathrm{p}(s_k, s_j)\mathrm{p}(s_k, X, i+1, d+1) \quad (15)$$

Then, the gradient for dynamic feature weight can be calculated similarly with equation 7 and 8:

$$\varphi_o = \left(\tilde{E}(h_o) - E(h_o)\right) \cdot \sigma^2 \quad (16)$$

In inference phase, Viterbi algorithm can still be applied to search the optimal path in the triangle lattice.

3.3. Dynamic feature

With the expanded framework, another important location reference besides current unit is available. Therefore, at least three categories of dynamic features can be supported:

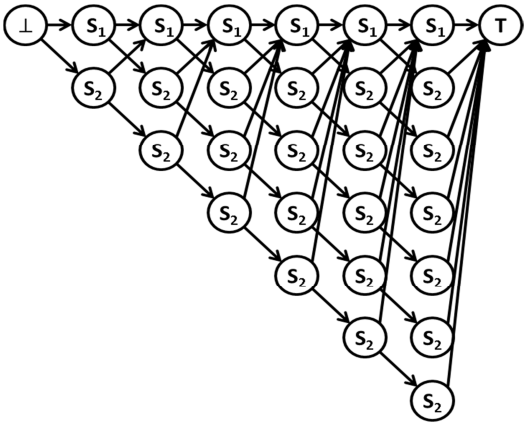

Figure 2 Expanded CRF

- Sum of a numeric feature in the range from previous key label to current unit.
  - Number of words from the previous L3.
  - Number of syllables from the previous L3.
- A feature around the previous key label.
  - POS of the word after previous L3.
  - Text of the word before previous L3.
- Any combination of above two categories of dynamic features and common unigram features.
  - Number of words from the previous L3 / Is Current word followed by punctuation.
  - POS of the word after previous L3 / POS of the next word.

4. Experiment

4.1. Corpus

Boston University Radio Speech Corpus (BURSC) [9] is selected for this study. It contains waveform recorded by professional radio announcers and scripts labeled with ToBI annotations [10]. The ToBI annotation consists of prosodic break, boundary tone and pitch accent. Hereinto, the prosodic break has six types from 0 to 5. The break 4 in that annotation means the intonation phrase break. For the experiments below, they are mapped into L1 and L3 as shown in Table 1. The original scripts are spitted into sentences and duplicates are removed. Finally, 978 sentences are obtained with 15188 L1 and 3718 L3 breaks.

Table 1 Break type mapping for BURSC

BURSC	Experiment
0,1,2,3	L1
4,5	L3

Also, we evaluate the dynamic features with an in-house corpus, named EUZ, which has larger size and wider domain coverage than BURSC. In total 5572 sentences, there are 55087 L1 and 9801 L3 breaks, manually annotated according to the recording by a professional speaker.

4.2. Feature template

Three feature templates are designed as below for experiment:

- Baseline

- Text
- POS
- Number of syllables in the word
- Is the word followed by punctuation
- Above features' combination
- Bigram feature

- Static length
  - All features in baseline
  - Number of words from previous punctuation
  - Number of words to next punctuation
  - Number of syllables from previous punctuation
  - Number of syllables to next punctuation
- Dynamic
  - All features in static length
  - Number of words from previous L3
  - Number of syllables from previous L3
  - POS of word after previous L3 / POS of current word
  - POS of word after previous L3 / POS of next word

4.3. Result and discussion

In evaluating the L3 break prediction model, 10-fold cross validation is implemented on each corpus. Performance is measured with three metrics: precision, recall and f-score. In fact, most punctuation marks are accompanied with L3 breaks and they are easily to be predicted. Therefore, the really challenging work is to predict those L3 breaks not at punctuations (see Table 3). Comparing the results in Table 2 and Table 3, the number drop is outstanding since the easy work occupies a substantial proportion. In BURSC, 1802 out of 3718 L3 breaks appear with punctuation marks, reaching almost half. For this reason, the results in Table 3 are more meaningful and the results of EUZ are only calculated in this way (see Table 4).

Table 2 L3 break prediction results of BURSC

L3	Precision (%)	Recall (%)	F-score (%)
Baseline	81.18	69.15	74.68
Static Length	80.69	70.36	75.17
Dynamic	80.47	72.27	76.15

Table 3 L3* break prediction results of BURSC

L3*	Precision (%)	Recall (%)	F-score (%)
Baseline	60.59	41.49	49.26
Static Length	60.61	43.84	50.88
Dynamic	61.33	47.60	53.60

***** excluding the L3 breaks at punctuations.

Results show that the recall is much less than the precision in all the cases. A phenomenon has generally been observed that CRF trends not to output the label type sparse in the training data. Accordingly, L3 breaks are predicted observably less than the expected. With the introduction of static length features, more L3 breaks are predicted and the recall increases remarkably. F-score achieves 3.29% and 26.27% relative growth on BURSC and EUZ respectively. Further, by employing the dynamic features, the relative F-score growth reaches 5.35% and 31.91% based on the static length.

As mentioned above, BURSC has small size and scripts are in narrow domain. That's why the baseline can obtain satisfied results with common features. Moreover, the intonation phrase length distribution learned from limited data is not reliable enough. For this reason, the improvement achieved by static length and dynamic features are less attractive. While for EUZ, a more general case, both word coverage and linguist context become more variable. In this case, baseline cannot behavior as good as on BURSC. The distance information plays a more important role and the improvement is accordingly more outstanding.

Table 4 L3* break prediction results of EUZ

L3*	Precision (%)	Recall (%)	F-score (%)
Baseline	48.06	18.25	26.46
Static Length	59.78	23.18	33.41
Dynamic	58.75	35.25	44.07

***** excluding the L3 breaks at punctuations.

5. Conclusion

This paper proposes an expanded CRF framework to hold location information of the previous L3 break. Based on it, various dynamic features can be supported to model long-distance dependencies.

Experiment results confirm the effect of distance information in L3 break prediction task. Also, expanded CRF framework with dynamic features shows its advantage in modeling such distance information. More generally, it may be said that its capability of molding long distance dependencies is verified. It is promising to apply and validate it in other tasks.

6. Acknowledgements

We thank Max Leung and Lei He for their support in this research. Also thank Yuanxiang Zhu and Zhiqiang Guo for their efforts in preparing the experiment materials.

7. References

[1] Lafferty, J., McCallum, A. and Pereira, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", Proc. of 18th International Conference Machine Learning, 282-289, 2001.

[2] Sun, J., Yang, J., Zhang, J. and Yan, Y., "Chinese prosody structure prediction based on Conditional Random Fields", Proc. of 5th International Conference on Natural Computation, 3:602-606, 2009.

[3] Qian, Y., Wu, Z., Ma, X. and Soong, F., "Automatic prosody prediction and detection with Conditional Random Field (CRF) models", ISCSLP, 7th International Symposium on, 135-138, 2010.

[4] Li, J., Hu, G. Wang, R. and Dai, L., "Sliding window smoothing for maximum entropy based intonational phrase prediction in Chinese", Proc. of 30th ICASSP, 285-288, 2005.

[5] Sarawagi, S. and Cohen, W.W., "Semi-Markov conditional random fields for information extraction", Advances in Neural Information Processing Systems, 17:1185–1192. MIT Press, Cambridge, MA, 2005.

[6] Sutton, C. and McCallum, A, "Collective segmentation and labeling of distant entities in information extraction", In Proc. of ICML'04 workshop on Statistical, 2004.

[7] Sutton, C., McCallum, A. and Rohanimanesh, K., "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data", Journal of Machine Learning Research, 8:693-723, 2007.

[8] Wallach, H.M., "Conditional Random Fields: An introduction", University of Pennsylvania CIS Technical Report MS-CIS-04-21, 2004.

[9] Ostendorf, M., Price, P.J. and Shattuck-Hufnagel S. "The Boston University radio news corpus".

[10] Ostendorf, M. and Ross, K., "A multi-level model for recognition of intonation labels", Computing Prosody, 291-308, 1997.