Dynamic Conditional Random Fields for Joint Sentence Boundary and Punctuation Prediction

Xuancong Wang[1], *Hwee Tou Ng*[1,2], *Khe Chai Sim*[2]

[1]NUS Graduate School for Integrative Sciences and Engineering
[2]Department of Computer Science, National University of Singapore

Abstract

The use of dynamic conditional random fields (DCRF) has been shown to outperform linear-chain conditional random fields (LCRF) for punctuation prediction on conversational speech texts [1]. In this paper, we combine lexical, prosodic, and modified n-gram score features into the DCRF framework for a joint sentence boundary and punctuation prediction task on TDT3 English broadcast news. We show that the joint prediction method outperforms the conventional two-stage method using LCRF or maximum entropy model (MaxEnt). We show the importance of various features using DCRF, LCRF, MaxEnt, and hidden-event n-gram model (HEN) respectively. In addition, we address the practical issue of feature explosion by introducing lexical pruning, which reduces model size and improves the F1-measure. We adopt incremental local training to overcome memory size limitation without incurring significant performance penalty. Our results show that adding prosodic and n-gram score features gives about 20% relative error reduction in all cases. Overall, DCRF gives the best accuracy, followed by LCRF, MaxEnt, and HEN.

Index Terms: punctuation, dynamic conditional random fields, sentence boundary detection

1. Introduction

The output from automatic speech recognition (ASR) does not contain punctuation marks, sentence boundaries, or word capitalization. Automatic sentence boundary detection and punctuation prediction are used to post-process ASR output for improved readability [2]. They also help to improve the accuracy of downstream natural language processing (NLP) applications such as machine translation.

Traditionally, sentence boundary detection and punctuation prediction have been treated as two separate tasks. In this paper, we focus on *joint* prediction of sentence boundaries and punctuation (in long utterances each containing multiple sentences) using DCRF on both correct recognition result (CRR) and ASR output. We compare the joint method with a conventional two-stage LCRF+LCRF or MaxEnt+MaxEnt method. We also address the practical issue of feature explosion. Similar to [3], we also show the contribution of prosodic features and modified language model (LM) features in addition to lexical features alone [1].

The rest of this paper is organized as follows. Section 2 introduces previous work in this area. Section 3 presents our approach integrating lexical, prosodic, and n-gram score features based on the framework of dynamic conditional random fields (DCRF). Section 4 describes feature extraction. The details of our experimental setup and results are presented in Section 5. Finally, Section 6 concludes this paper.

2. Previous Work

Much research on punctuation prediction has been carried out in speech and language processing. The earliest research exploited lexical features only. [4] used trigram language modeling for comma prediction by treating commas as words. [5] proposed a hidden event language model that treated sentence boundary detection and punctuation insertion as interword hidden event detection tasks. Their proposed method was implemented in the open-source utility hidden-ngram as part of the SRILM toolkit [6]. [7] presented a purely n-gram based approach using finite state automata (FSA) that jointly predicted punctuation and case information for English. The latest work by [1] made use of dynamic conditional random fields (DCRF) [8] by jointly predicting sentence type and punctuation, but without using prosodic or language model information.

Prosodic information has been shown to be helpful for punctuation prediction. There are several works making use of both prosodic and lexical features. [3] combined prosodic and lexical information for punctuation prediction. In their work, prosodic features were incorporated using classification and regression tree (CART), and lexical information was extracted in the form of language model scores. [9] investigated both finite state and multi-layer perceptron methods on punctuating broadcast news, making use of both prosodic and linguistic information. [10] used maximum entropy model (MaxEnt) for punctuation insertion in English conversational speech. [11] has compared MaxEnt, conditional random fields (CRF) and hidden Markov models (HMM) for sentence boundary detection and disfluency detection. In general, discriminative models like MaxEnt and CRF significantly outperform generative models like HMM and HEN.

3. Model Description

Maximum entropy models and conditional random fields are special cases of log-linear models. Their probabilities are expressed in a similar way as multinomial logistic regression. Let X_i denote the feature vector for the i^{th} sample and β_j denote the weight vector for the j^{th} class. In a maximum entropy model, the posterior of any class k is given by Eqn 1:

$$\Pr(y_i = k) = \frac{\exp(X_i \cdot \beta_k)}{\sum_{j=0}^{J} \exp(X_i \cdot \beta_j)} \tag{1}$$

Conditional random fields (CRF) extend MaxEnt by modeling the joint distribution of adjacent output variables, in the form of an undirected graphical model. The relationship among different variables can thus be captured in CRF. This turns out to be very useful in many applications because the prediction at each time instance is no longer in isolation from one another.

In linear-chain CRF (LCRF), the graph is in the form of a one-dimensional chain of variables.

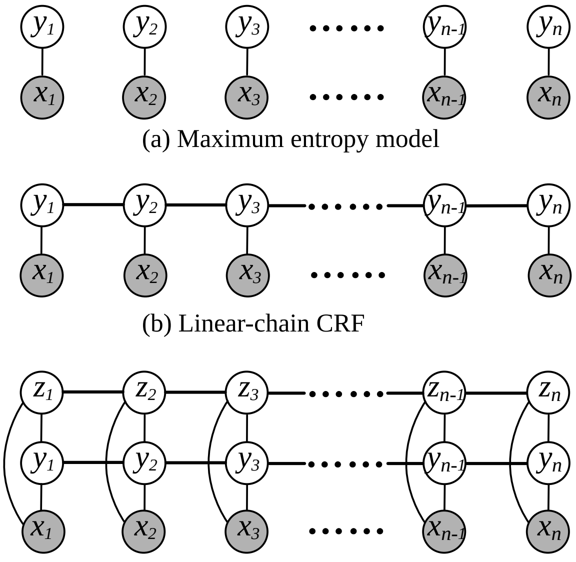
Formally, a first-order linear-chain CRF which assumes the first-order Markov property defines the conditional probability of a label sequence vector **y** given input **x** as in Eqn. 2:

$$P_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_t \sum_m \lambda_m f_m(\mathbf{x}, y_t, y_{t+1}, t)\right) \quad (2)$$

where f_m are the edge feature functions; $Z(\mathbf{x})$ is the normalizing factor to ensure the conditional probabilities over all possible outcomes sum up to 1.

Dynamic CRF (DCRF) extends LCRF by allowing the graph to have any arbitrary structure. For language and speech processing, the input that we want to label is often in the form of a linear sequence, so DCRF needs to have repetitive structures to cover each node of the input sequence.

Factorial CRF (FCRF) is a special case of DCRF. It connects variables in different layers by introducing a pairwise factor at each time index to form a rectangular lattice structure. It can capture the joint distribution of various layers so that the prediction of any layer is not in isolation from the other layers. An illustration of these different types of undirected graphical models is shown in Fig. 1. Shaded nodes are observed nodes.



(a) Maximum entropy model

(b) Linear-chain CRF

(c) 2-layer factorial CRF

Figure 1: A graphical representation of the three basic undirected graphical models. y_i denotes the 1st layer label, z_i denotes the 2nd layer label, and x_i denotes the observation sequence

Formally, a dynamic CRF is defined by Eqn. 3 as follows:

$$P_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\sum_t \sum_{c \in \mathcal{C}} \sum_k \lambda_k f_k(\mathbf{x}, y_{(c,t)}, t)) \quad (3)$$

where $\mathcal{C}$ is a set of clique indices, and $y_{(c,t)}$ is the set of variables in the unrolled graph at clique index c and time index t [8]. A clique refers to any node (first order clique) or edge (second order clique) in the graph. For our task, only edge cliques are used and adding node cliques does not help. Thus, the DCRF model contains about three times the total number of parameters as the LCRF model due to the two additional cliques.

We consider punctuation prediction as a sequence labeling problem in which the punctuation marks are the labels. A two-layer factorial CRF predicts two layers of labels, one layer for punctuation marks, and the other layer for sentence tags. For the first layer, we consider the following 5 punctuation tags: Comma, Period, QMark (question mark), EMark (exclamation mark), and None (no punctuation). For the second layer, we consider the following 6 sentence tags: DeBeg, DeIn, QnBeg, QnIn, ExBeg, and ExIn. Here, “De”, “Qn”, and “Ex” stand for declarative, question, and exclamatory sentences respectively; “Beg” and “In” denote the beginning and non-beginning of a sentence respectively. An example is shown in Fig. 2.

4. Feature Extraction

In our experiments, we consider 3 classes of features: lexical features, prosodic features, and normalized 3-gram language model score features.

4.1. Lexical features

We consider all 1-grams, 2-grams, and 3-grams within 5 words of the current word as binary feature functions. For example, in the sentence in Fig. 2, the lexical features at the fourth word ‘been’ are:
1-gram features: ⟨–2⟩@–5 ⟨–1⟩@–4 ... NOW@+5
2-gram features: ⟨–2⟩+⟨–1⟩@–5 ... YEARS+NOW@+4
3-gram features: ⟨–2⟩+⟨–1⟩+GOSH@–5 ...

Vocabulary pruning is applied by finding the top V (3,000 in our case) most frequent words in the training data and mapping the rest to ⟨UNK⟩ before lexical feature extraction. We observe that as the vocabulary size is reduced, the F-measure increases slightly first and then decreases (Fig. 3). The maximum point occurs when the vocabulary size is about 25% of the total size (11,510). There are 2 reasons for this. Firstly, since the model is sparse, pruning the vocabulary decreases the model sparsity, and the training process tends to find a better local optimum with reduced model sparsity. Secondly, mapping rare words to ⟨UNK⟩ reduces the feature miss rate. For example, suppose the 3-gram feature “am-a-student” occurs in the training data and “am-a-pilot” occurs in the test data but not in the training data and thus this feature will be missed if no vocabulary pruning is done. However, if both ‘student’ and ‘pilot’ are mapped to ⟨UNK⟩, this feature “am-a-⟨UNK⟩” will not be missed. In other words, knowing that a word is rare (i.e., ⟨UNK⟩) does better than knowing nothing at all.

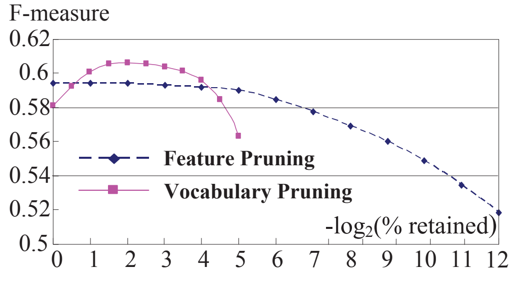


Figure 3: The effect of vocabulary pruning and feature pruning

4.2. Prosodic features

Prosodic features are acoustic cues such as pitch, loudness, etc. extracted from the raw speech audio. On this task, we use pause duration, average F0 ratio, and energy ratio, similar to

Second layer:	QnBeg	QnIn	QnIn	QnIn	QnIn	QnIn	QnIn	QnIn	QnIn	DeBeg	DeIn	
First layer:	Comma	None	None	None	None	None	None	None	QMark	None	Period	
...	Gosh	so	she's	been	underground	for	twelve	years	now	That's	right	...

Reference: *Gosh, so she's been underground for twelve years now? That's right.*

Figure 2: An example showing the two layers of factorial CRF labels for a sentence in TDT3 English corpus

[3]. Phoneme boundaries are determined using Viterbi forced alignment, and F0 is obtained using Praat [12]. Continuous values are quantized into 8 discrete bins to give rise to discrete features for DCRF.

4.3. Normalized n-gram language model scores

A 3-gram language model is trained on the combined Gigaword corpus [13] TDT2 and TDT3 [14] data (excluding the test data). Comma and sentence-end punctuation are treated as distinct words. The two LM features are defined in Eqn. 4:

$$\begin{aligned} \text{Normalized n-gram score 1} &= \frac{P(\langle\text{/s}\rangle|w_{-1}, w_0)}{P(w_{+1}|w_{-1}, w_0)} \\ \text{Normalized n-gram score 2} &= \frac{P(\langle\text{Comma}\rangle|w_{-1}, w_0)}{P(w_{+1}|w_{-1}, w_0)} \end{aligned} \tag{4}$$

where $\langle$/s$\rangle$ is the sentence-end tag, and w_{-1}, w_0, and w_{+1} refer to the preceding word, the current word, and the next word respectively. The normalized n-gram score measures the ratio in the likelihood of a punctuation occurring against not occurring. The intuition is that in regions where a punctuation mark should occur, the n-gram probability of $\langle$/s$\rangle$ or $\langle$Comma$\rangle$ is on average higher. The normalized n-gram scores outperform raw n-gram scores used in previous works because it is more discriminative with respect to the local context. The limitation of raw n-gram scores is that the magnitudes of raw scores are with respect to the global context only, which may or may not fit into the local context which also depends on the next word.

5. Experiments

5.1. Data preparation

We selected our data from the two TDT3 broadcast news corpora. Only the news sections of the two radio sources (PRI and VOA_ENG) are used. Text normalization is done to convert numbers, dates, etc. Sentences containing punctuation marks other than periods, commas, question marks, and exclamation marks are discarded. Untranscribed segments and music segments are also discarded. The original texts contain too few question marks (about 1%) and exclamation marks (<0.1%). In order to balance the different punctuation marks, we gave priority to sentences containing question or exclamation marks when selecting stories. After preprocessing, there are in total about 150,000 word tokens, 17,251 punctuation marks (7,502 periods, 8,351 commas, 1,386 question marks, and 12 exclamation marks), and 1,679 instances. Finally, our preprocessed data corpus is randomly split into two parts, 90% for training and 10% for testing.

5.2. Incremental local training

Given limited memory, the total number of features is too large to allow global training using all the features. Thus, we propose the following incremental local training scheme:

1. Sort all features in descending order of frequency.
2. Select the top N (depending on memory size, 800,000 in our experiments) most frequent features and perform local training using only these N features and employing L1-norm regularization.
3. Prune away features with the smallest absolute weights, retaining $N \cdot r$ features, where $1 - r$ is the pruning ratio.
4. Add more features until N features are chosen.
5. Perform local training using the selected N features.
6. Goto step 3 if not all features have been covered.

We have run a comparison test between global (all 0.9M features) and local training ($N = 0.1M$) on the evaluation data with $V = 1,000$. We choose $r = 0.5$. The degradation in accuracy is insignificant ($\Delta F1 < 0.1$) because L1-norm regularization identifies useful (important) features and assigns to them non-zero (larger) weights. The effect of feature pruning is presented in Fig. 3, which shows that the model is sparse and the degradation is negligible as long as more than 5% of the total features are kept. This also gives us an estimated minimum value of $N \cdot r$ below which incremental local training will start to incur significant degradation.

5.3. Experimental results

We have applied the hidden event n-gram (5-gram) model, MaxEnt, LCRF, and DCRF on the same set of preprocessed training and test data using the same features. Prediction accuracy is measured by precision, recall, and F1 measure:

$$\begin{aligned} \text{Precision} &= \frac{\text{\# Correctly predicted punctuation symbols}}{\text{\# predicted punctuation symbols}} \\ \text{Recall} &= \frac{\text{\# Correctly predicted punctuation symbols}}{\text{\# expected punctuation symbols}} \\ \text{F1 measure} &= \frac{2}{\text{1/Precision+1/Recall}} \end{aligned}$$

For ASR, the word error rate of the speech recognizer is 34.6%. A prediction is considered correct if the predicted punctuation mark is the same as the reference and its time interval overlaps with the reference.

We have used SRILM for the hidden event n-gram model, Mallet GRMM package (with modification) for the LCRF and DCRF model. We have used tree-based re-parametrization (TRP) for the inference algorithm and L-BFGS for gradient search. The results are shown in Table 1. The F1 measure for the hidden-event n-gram model (using lexical features only) are 31.6% for CRR and 19.4% for ASR output, much lower than the discriminative models.

The results clearly show that adding prosodic features improves punctuation prediction accuracy significantly in all cases. And the general performance trend of the various models is that DCRF > LCRF > MaxEnt > hidden event n-gram, using different feature combinations. Another important observation is that on reference transcripts, lexical features alone perform much better than prosodic features, while on ASR output, this difference is reduced significantly. This is because prosodic features are derived directly from the raw speech signal and are thus more tolerant to ASR errors. However, ASR errors may also result in alignment errors of the word boundaries which may in turn corrupt the prosodic features. Note that our DCRF baseline on CRR using only lexical features (F1 = 48.3%) is much lower than that of [1] (F1 = 88.6%), since punctuation prediction accuracy is much lower when sentences are longer and each instance contains more words and punctuation marks. The average number of punctuation marks per instance is 10.4 in our data and 1.4 in the IWSLT09 data used by [1].

Features	MaxEnt		LCRF		DCRF	
Input	CRR	ASR	CRR	ASR	CRR	ASR
Lex	44.2	25.2	45.6	24.5	48.3	26.8
Pro	32.5	23.6	33.3	24.3	33.4	25.5
LM	19.8	3.1	21.5	3.8	22.6	5.2
Lex+Pro	51.8	35.4	55.0	37.5	57.1	39.0
Lex+LM	49.8	29.4	51.8	31.7	54.3	33.9
Pro+LM	42.2	31.7	41.8	31.3	43.9	32.4
All	55.6	38.5	58.4	41.2	**61.4**	**42.8**

Table 1: Comparison of punctuation prediction F1 measures (in %) using different algorithms and features.

5.4. Comparison to a two-stage LCRF+LCRF

In order to show that the DCRF joint prediction method indeed outperforms a two-stage method both using LCRF, we have trained two LCRF models. The first model is for sentence tag prediction using all features described in Section 4. The second model is trained for punctuation prediction using the same features (lexical features beyond sentence boundary are updated) with the sentence tag as an additional feature. Moreover, it is trained on correctly split sentences (each training instance containing only one sentence) because punctuation prediction is much more accurate on short utterances. Sentence tag is first predicted using the first LCRF model and sentences are split based on derived sentence boundaries. Then the second LCRF model is applied on each of the predicted sentence segments with the predicted sentence tag as an additional feature.

Input	MaxEnt	LCRF	DCRF
CRR	74.6	77.1	78.2
ASR	54.7	56.6	57.8

Table 2: Comparison of F1 measures (in %) on sentence boundary detection (Stage 1)

Input	MaxEnt	LCRF	DCRF
CRR	56.1	59.9	61.4
ASR	38.9	41.9	42.8

Table 3: Comparison of F1 measures (in %) on punctuation prediction (Stage 2) (using the predicted sentence boundaries from Stage 1 for LCRF and MaxEnt)

All differences are statistically significant. From the results, it can be seen that the two-stage method performs slightly better than one-stage punctuation baseline in Table 1 for LCRF and MaxEnt. However, it is significantly worse than DCRF punctuation prediction due to error propagation from the first layer to the second layer. In particular, DCRF joint prediction is better in terms of both sentence boundary and punctuation prediction.

6. Conclusion

In this paper, we have combined prosodic features, normalized n-gram score features, and lexical features into the DCRF framework for joint sentence boundary and punctuation prediction on long text segments. In particular, adding prosodic and n-gram score features in addition to lexical features reduces the prediction error $(1 - F1)$ by about 20% relatively. We experimentally show that DCRF joint prediction significantly outperforms two-stage LCRF/MaxEnt sentence boundary prediction and LCRF/MaxEnt punctuation prediction on split sentences in both stages. In all cases, DCRF gives the best result. Moreover, keeping the top 25% most frequent words and mapping the rest to ⟨UNK⟩ gives the optimal performance. To achieve scalability, incremental local training can be adopted without incurring significant performance degradation.

7. Acknowledgment

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

8. References

[1] W. Lu and H. T. Ng, “Better punctuation prediction with dynamic conditional random fields,” in *EMNLP 2010*.

[2] M. Ostendorf, B. Favre, R. Grishman, D. Hakkani-Tur, M. Harper, D. Hillard, J. Hirschberg, H. Ji, J. Kahn, Y. Liu *et al.*, “Speech segmentation and spoken document processing,” *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 59–69, 2008.

[3] J. Kim and P. Woodland, “The use of prosody in a combined system for punctuation generation and speech recognition,” in *Europ. Conf. on Speech Comm. and Tech. 2001*.

[4] D. Beeferman, A. Berger, and J. Lafferty, “Cyberpunc: A lightweight punctuation annotation system for speech,” in *ICASSP 1998*.

[5] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu, “Automatic detection of sentence boundaries and disfluencies based on recognized words,” in *ICSLP 1998*.

[6] A. Stolcke, “SRILM – An extensible language modeling toolkit,” in *ICSLP 2002*.

[7] A. Gravano, M. Jansche, and M. Bacchiani, “Restoring punctuation and capitalization in transcribed speech,” in *ICASSP 2009*.

[8] C. Sutton, K. Rohanimanesh, and A. McCallum, “Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data,” in *ICML 2004*.

[9] H. Christensen, Y. Gotoh, and S. Renals, “Punctuation annotation using statistical prosody models,” in *ISCA Workshop on Prosody in Speech Recognition and Understanding, 2001*.

[10] J. Huang and G. Zweig, “Maximum entropy model for punctuation annotation from speech,” in *Eurospeech 2002*.

[11] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, “Enriching speech recognition with automatic detection of sentence boundaries and disfluencies,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1526–1540, 2006.

[12] P. Boersma and D. Weenink, “Praat, a system for doing phonetics by computer,” *Glot International*, vol. 5, pp. 341–345, 2001.

[13] D. Graff, J. Kong, K. Chen, and K. Maeda, “English Gigaword,” *Catalog number LDC2003T05, Linguistic Data Consortium, Philadelphia*, 2003.

[14] C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel, “Large multilingual broadcast news corpora for cooperative research in topic detection and tracking: The TDT2 and TDT3 corpus efforts,” in *Proceedings of Language Resources and Evaluation Conference, Athens, Greece*, 2000.