Better Punctuation Prediction with Dynamic Conditional Random Fields

Wei Lu and **Hwee Tou Ng**
Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore 117417
{luwei,nght}@comp.nus.edu.sg

Abstract

This paper focuses on the task of inserting punctuation symbols into transcribed conversational speech texts, without relying on prosodic cues. We investigate limitations associated with previous methods, and propose a novel approach based on dynamic conditional random fields. Different from previous work, our proposed approach is designed to jointly perform both sentence boundary and sentence type prediction, and punctuation prediction on speech utterances.

We performed evaluations on a transcribed conversational speech domain consisting of both English and Chinese texts. Empirical results show that our method outperforms an approach based on linear-chain conditional random fields and other previous approaches.

1 Introduction

Outputs of standard automatic speech recognition (ASR) systems typically consist of utterances where important linguistic and structural information (e.g., true case, sentence boundaries, punctuation symbols, etc) is not available. Such information is crucial in improving the readability of the transcribed speech texts, and plays an important role when further processing is required, such as in part-of-speech (POS) tagging, parsing, information extraction, and machine translation.

We focus on the punctuation prediction task in this work. Most previous punctuation prediction techniques, developed mostly by the speech processing community, exploit both lexical and prosodic cues. However, in order to fully exploit prosodic features such as pitch and pause duration, it is necessary to have access to the original raw speech waveforms. In some scenarios where further natural language processing (NLP) tasks on the transcribed speech texts become the main concern, speech prosody information may not be readily available. For example, in the recent evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) (Paul, 2009), only manually transcribed or automatically recognized speech texts are provided but the original raw speech waveforms are not available.

In this paper, we tackle the task of predicting punctuation symbols from a standard text processing perspective, where only the speech texts are available, without relying on additional prosodic features such as pitch and pause duration. Specifically, we perform the punctuation prediction task on transcribed conversational speech texts, using the IWSLT corpus (Paul, 2009) as the evaluation data.

Different from many other corpora such as broadcast news corpora, a conversational speech corpus consists of dialogs where informal and short sentences frequently appear. In addition, due to the nature of conversation, it also contains more question sentences compared to other corpora. An example English utterance randomly selected from the IWSLT corpus, along with its punctuated and cased version, are shown below:

you are quite welcome and by the way we may get other reservations so could you please call us as soon as you fix the date

You are quite welcome . And by the way , we may get other reservations , so could you please call us as soon as you fix the date ?

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 177–186,
MIT, Massachusetts, USA, 9-11 October 2010.

The rest of this paper is organized as follows. We start with surveying related work in Section 2. One class of widely-used previous techniques is then studied in detail in Section 3. Next, we investigate methods for improving existing methods in Section 4 and 5. Empirical evaluation results are presented and discussed in Section 6. We finally conclude in Section 7.

2 Related Work

Punctuation prediction has been extensively studied in the speech processing field. It is also sometimes studied together with a closely related task – sentence boundary detection.

Much previous work assumes that both lexical and prosodic cues are available for the task. Kim and Woodland (2001) performed punctuation insertion during speech recognition. Prosodic features together with language model probabilities were used within a decision tree framework. Christensen et al. (2001) focused on the broadcast news domain and investigated both finite state and multi-layer perceptron methods for the task, where prosodic and lexical information was incorporated. Huang and Zweig (2002) presented a maximum entropy-based tagging approach to punctuation insertion in spontaneous English conversational speech, where both lexical and prosodic features were exploited. Liu et al. (2005) focused on the sentence boundary detection task, by making use of conditional random fields (CRF) (Lafferty et al., 2001). Their method was shown to improve over a previous method based on hidden Markov model (HMM).

There is relatively less work that exploited lexical features only. Beeferman et al. (1998) focused on comma prediction with a trigram language model. A joint language model was learned from punctuated texts, and commas were inserted so as to maximize the joint probability score. Recent work by Gravano et al. (2009) presented a purely n-gram based approach that jointly predicted punctuation and case information of English.

Stolcke et al. (1998) presented a "hidden event language model" that treated boundary detection and punctuation insertion as an interword hidden event detection task. Their proposed method was implemented in the handy utility *hidden-ngram* as part of the SRILM toolkit (Stolcke, 2002). It was widely used in many recent spoken language translation tasks as either a preprocessing (Wang et al., 2008) or postprocessing (Kirchhoff and Yang, 2007) step. More details about this model will be given in the next section.

Recently, there are also several research efforts that try to optimize some downstream application after punctuation prediction, rather than the prediction task itself. Examples of such downstream applications include punctuation prediction for part-of-speech (POS) tagging and name tagging (Hillard et al., 2006), statistical machine translation (Matusov et al., 2006), and information extraction (Favre et al., 2008).

3 Hidden Event Language Model

Many previous research efforts consider the boundary detection and punctuation insertion task as a hidden event detection task. One such well-known approach was introduced by Stolcke et al. (1998). They adopted a HMM to describe a joint distribution over words and interword events, where the observations are the words, and the word/event pairs are encoded as hidden states. Specifically, in this task word boundaries and punctuation symbols are encoded as interword events. The training phase involves training an n-gram language model over all observed words and events with smoothing techniques. The learned n-gram probability scores are then used as the HMM state-transition scores. During testing, the posterior probability of an event at each word is computed with dynamic programming using the forward-backward algorithm. The sequence of most probable states thus forms the output which gives the punctuated sentence.

Such a HMM-based approach has several drawbacks. First, the n-gram language model is only able to capture surrounding contextual information. However, we argue that in many cases, modeling of longer range dependencies is required for punctuation insertion. For example, the method is unable to effectively capture the long range dependency between the initial phrase "*would you*" which strongly indicates a question sentence, and an ending question mark. This hurts the punctuation prediction performance for our task since we are particularly interested in conversational speech texts where question sentences appear frequently.

Thus, in practice, special techniques are usually required on top of using a hidden event language model in order to overcome long range dependencies. Examples include relocating or duplicating punctuation symbols to different positions of a sentence such that they appear closer to the indicative words (e.g., "*how much*" indicates a question sentence). One such technique was introduced by the organizers of the IWSLT evaluation campaign, who suggested duplicating the ending punctuation symbol to the beginning of each sentence before training the language model[1]. Empirically, the technique has demonstrated its effectiveness in predicting question marks in English, since most of the indicative words for English question sentences appear at the beginning of a question. However, such a technique is specially designed and may not be widely applicable in general or to languages other than English. Furthermore, a direct application of such a method may fail in the event of multiple sentences per utterance without clearly annotated sentence boundaries within an utterance.

Another drawback associated with such an approach is that the method encodes strong dependency assumptions between the punctuation symbol to be inserted and its surrounding words. Thus, it lacks the robustness to handle cases where noisy or out-of-vocabulary (OOV) words frequently appear, such as in texts automatically recognized by ASR systems. In this paper, we devise techniques based on conditional random fields to tackle the difficulties due to long range dependencies.

4 Linear-Chain Conditional Random Fields

One natural approach to relax the strong dependency assumptions encoded by the hidden event language model is to adopt an undirected graphical model, where arbitrary overlapping features can be exploited.

Conditional random fields (CRF) (Lafferty et al., 2001) have been widely used in various sequence labeling and segmentation tasks (Sha and Pereira, 2003; Tseng et al., 2005). Unlike a HMM which models the joint distribution of both the label sequence and the observation, a CRF is a discriminative model of the conditional distribution of the complete label sequence given the observation.

Specifically, a first-order linear-chain CRF which assumes first-order Markov property is defined by the following equation:

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_t \sum_k \lambda_k f_k(\mathbf{x}, y_{t-1}, y_t, t)\right) \quad (1)$$

where $\mathbf{x}$ is the observation and $\mathbf{y}$ is the label sequence. Feature functions f_k with time step t are defined over the entire observation $\mathbf{x}$ and two adjacent hidden labels. $Z(\mathbf{x})$ is a normalization factor to ensure a well-formed probability distribution. Figure 1 gives a simplified graphical representation of the model, where only the dependencies between label and observation in the same time step are shown.

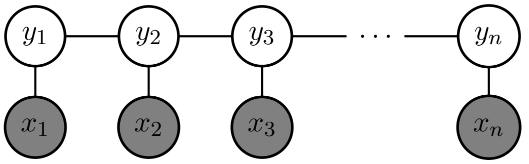

Figure 1: A simplified graphical representation for linear-chain CRF (observations are shaded)

proposed tags
NONE COMMA (,) PERIOD (.)
QMARK (?) EMARK (!)

Table 1: The set of all possible tags for linear-chain CRF

We can model the punctuation prediction task as the process of assigning a tag to each word, where the set of possible tags is given in Table 1. That is, we assume each word can be associated with an event, which tells us which punctuation symbol (possibly NONE) should be inserted after the word. The training data consists of a set of utterances where punctuation symbols are encoded as tags that are assigned to the individual words. The tag NONE means no punctuation symbol is inserted after the current word. Any other tag refers to inserting the corresponding punctuation symbol. In the testing phase, the most probable sequence of tags is

[1] http://mastarpj.nict.go.jp/IWSLT2008/downloads/case+punc_tool_using_SRILM.instructions.txt

Sentence: *no , please do not . would you save your questions for the end of my talk , when i ask for them ?*

no	*please*	*do*	*not*	*would*	*you*	...	*my*	*talk*	*when*	...	*them*
COMMA	NONE	NONE	PERIOD	NONE	NONE	...	NONE	COMMA	NONE	...	QMARK

Figure 2: An example tagging of a training sentence for the linear-chain CRF

predicted and the punctuated text can then be constructed from such an output. An example tagging of an utterance is illustrated in Figure 2.

Following (Sutton et al., 2007), we factorize a feature of conditional random fields as a product of a binary function on assignment of the set of cliques at the current time step (in this case an edge), and a feature function solely defined on the observation sequence. n-gram occurrences surrounding the current word, together with position information, are used as binary feature functions, for $n = 1, 2, 3$. All words that appear within 5 words from the current word are considered when building the features. Special start and end symbols are used beyond the utterance boundaries. For example, for the word *do* shown in Figure 2, example features include unigram features *do@0*, *please@-1*, bigram feature *would+you@[2,3]*, and trigram feature *no+please+do@[-2,0]*.

Such a linear-chain CRF model is capable of modeling dependencies between words and punctuation symbols with arbitrary overlapping features, thus avoiding the strong dependency assumptions in the hidden event language model. However, the linear-chain CRF model still exhibits several problems for the punctuation task. In particular, the dependency between the punctuation symbols and the indicative words cannot be captured adequately, if they appear too far away from each other. For example, in the sample utterance shown in Figure 2, the long range dependency between the ending question mark and the indicative words *would you* which appear very far away cannot be directly captured. The problem arises because a linear-chain CRF only learns a sequence of tags at the individual word level but is not fully aware of sentence level information, such as the start and end of a complete sentence.

Hence, it would be more reasonable to hypothesize that the punctuation symbols are annotated at the sentence level, rather than relying on a limited window of surrounding words. A model that can jointly perform sentence segmentation and sentence type prediction, together with word level punctuation prediction would be more beneficial for our task. This motivates us to build a joint model for performing such a task, to be presented in the next section.

5 Factorial Conditional Random Fields

Extensions to the linear-chain CRF model have been proposed in previous research efforts to encode long range dependencies. One such well-known extension is the semi-Markov CRF (semi-CRF) (Sarawagi and Cohen, 2005). Motivated by the hidden semi-Markov model, the semi-CRF is particularly helpful in text chunking tasks as it allows a state to persist for a certain interval of time steps. This in practice often leads to better modeling capability of chunks, since state transitions within a chunk need not precisely follow the Markov property as in the case of linear-chain CRF. However, it is not clear how such a model can benefit our task, which requires word-level labeling in addition to sentence boundary detection and sentence type prediction.

The skip-chain CRF (Sutton and McCallum, 2004), another variant of linear-chain CRF, attaches additional edges on top of a linear-chain CRF for better modeling of long range dependencies between states with similar observations. However, such a model usually requires known long range dependencies in advance and may not be readily applicable to our task where such clues are not explicit.

As we have discussed above, since we would like to jointly model both the word-level labeling task and the sentence-level annotation task (sentence boundary detection and sentence type prediction), introducing an additional layer of tags to perform both tasks together would be desirable. In this section, we propose the use of factorial CRF (F-CRF) (Sutton et al., 2007), which has previously been shown to be effective for joint labeling of multiple sequences (McCallum et al., 2003).

The F-CRF as a specific case of dynamic conditional random fields was originally motivated from dynamic Bayesian networks, where an identical structure repeats over different time steps. Analogous to the linear-chain CRF, one can think of the F-CRF as a framework that provides the capability of simultaneously labeling multiple layers of tags for a given sequence. It learns a joint conditional distribution of the tags given the observation. Formally, dynamic conditional random fields define the conditional probability of a sequence of label vectors **y** given the observation **x** as:

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_t \sum_{c\in\mathcal{C}} \sum_k \lambda_k f_k(\mathbf{x}, y_{(c,t)}, t)\right) \tag{2}$$

where cliques are indexed at each time step, $\mathcal{C}$ is a set of clique indices, and $y_{(c,t)}$ is the set of variables in the unrolled version of a clique with index c at time t (Sutton et al., 2007). Figure 3 gives a graphical representation of a two-layer factorial CRF, where the cliques include the two within-chain edges (e.g., $z_2 - z_3$ and $y_2 - y_3$) and one between-chain edge (e.g., $z_3 - y_3$) at each time step.

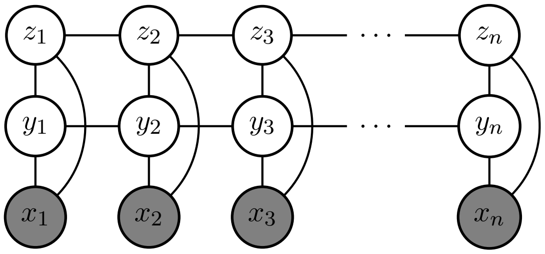

Figure 3: A two-layer factorial CRF

layer	proposed tags
word	NONE,COMMA,PERIOD, QMARK,EMARK
sentence	DEBEG,DEIN,QNBEG,QNIN, EXBEG,EXIN

Table 2: The set of all possible tags proposed for each layer

We build two layers of labels for this task, as listed in Table 2. The word layer tags are responsible for inserting a punctuation symbol (including NONE) after each word, while the sentence layer tags are used for annotating sentence boundaries and identifying the sentence type (declarative, question, or exclamatory). Tags from the word layer are the same as those of the linear-chain CRF. The sentence layer tags are designed for three types of sentences. DEBEG and DEIN indicate the start and the inner part of a declarative sentence respectively, likewise for QNBEG and QNIN (question sentences), as well as EXBEG and EXIN (exclamatory sentences). The same example utterance we looked at in the previous section is now tagged with these two layers of tags, as shown in Figure 4. Analogous feature factorization and the same n-gram feature functions used in linear-chain CRF are used in F-CRF.

When learning the sentence layer tags together with the word layer tags, the F-CRF model is capable of leveraging useful clues learned from the sentence layer about sentence type (e.g., a question sentence, annotated with QNBEG, QNIN, QNIN, ..., or a declarative sentence, annotated with DEBEG, DEIN, DEIN, ...), which can be used to guide the prediction of the punctuation symbol at each word, hence improving the performance at the word layer. For example, consider jointly labeling the utterance shown in Figure 4. Intuitively, when evidences show that the utterance consists of two sentences – a declarative sentence followed by a question sentence, the model tends to annotate the second half of the utterance with the sequence QNBEG QNIN .... This in turn helps to predict the word level tag at the end of the utterance as QMARK, given the dependencies between the two layers existing at each time step. In practice, during the learning process, the two layers of tags are jointly learned, thus providing evidences that influence each other's tagging process.

In this work, we use the GRMM package (Sutton, 2006) for building both the linear-chain CRF (L-CRF) and factorial CRF (F-CRF). The tree-based reparameterization (TRP) schedule for belief propagation (Wainwright et al., 2001) is used for approximate inference.

6 Experiments

We perform experiments on part of the corpus of the IWSLT09 evaluation campaign (Paul, 2009), where both Chinese and English conversational speech

Sentence: *no , please do not . would you save your questions for the end of my talk , when i ask for them ?*

no	*please*	*do*	*not*	*would*	*you*	...	*my*	*talk*	*when*	...	*them*
Comma	None	None	Period	None	None	...	None	Comma	None	...	Qmark
DeBeg	DeIn	DeIn	DeIn	QnBeg	QnIn	...	QnIn	QnIn	QnIn	...	QnIn

Figure 4: An example tagging of a training sentence for the factorial CRF

texts are used. Two multilingual datasets are considered, the BTEC (Basic Travel Expression Corpus) dataset and the CT (Challenge Task) dataset. The former consists of tourism-related sentences, and the latter consists of human-mediated cross-lingual dialogs in travel domain. The official IWSLT09 BTEC training set consists of 19,972 Chinese-English utterance pairs, and the CT training set consists of 10,061 such pairs. We randomly split each of the two datasets into two portions, where 90% of the utterances are used for training the punctuation prediction models, and the remaining 10% for evaluating the prediction performance. For all the experiments, we use the default segmentation of Chinese as provided, and English texts are preprocessed with the Penn Treebank tokenizer[2]. We list the statistics of the two datasets after processing in Table 3. The proportions of sentence types in the two datasets are listed. The majority of the sentences are declarative sentences. However, question sentences are more frequent in the BTEC dataset compared to the CT dataset. Exclamatory sentences contribute less than 1% for all datasets and are not listed. We also count how often each utterance consists of multiple sentences. The utterances from the CT dataset are much longer (with more words per utterance), and therefore more CT utterances actually consist of multiple sentences.

	BTEC		CT	
	Cn	En	Cn	En
declarative sent.	64%	65%	77%	81%
question sent.	36%	35%	22%	19%
multi.sent./uttr.	14%	17%	29%	39%
avg.words./uttr.	8.59	9.46	10.18	14.33

Table 3: Statistics of the BTEC and CT datasets

For the methods based on the hidden event language model, we design extensive experiments due to many possible setups. Specifically, these experiments can be divided into two categories: with or without duplicating the ending punctuation symbol to the start of a sentence before training. This setting can be used to assess the impact of the proximity between the punctuation symbol and the indicative words for the prediction task. Under each category, two possible approaches are tried. The single pass approach performs prediction in one single step, where all the punctuation symbols are predicted sequentially from left to right. In the cascaded approach, we format the training sentences by replacing all sentence-ending punctuation symbols with special sentence boundary symbols first. A model for sentence boundary prediction is learned based on such training data. This step is then followed by predicting the actual punctuation symbols. Both trigram and 5-gram language models are tried for all combinations of the above settings. This gives us a total of 8 possible combinations based on the hidden event language model. When training all the language models, modified Kneser-Ney smoothing (Chen and Goodman, 1996) for n-grams is used.

To assess the performance of the punctuation prediction task, we compute precision ($prec.$), recall ($rec.$), and F1-measure (F_1), as defined by the following equations:

$$\begin{aligned} prec. &= \frac{\text{\# Correctly predicted punctuation symbols}}{\text{\# predicted punctuation symbols}} \\ rec. &= \frac{\text{\# Correctly predicted punctuation symbols}}{\text{\# expected punctuation symbols}} \\ F_1 &= \frac{2}{1/prec. + 1/rec.} \end{aligned}$$

6.1 Performance on Correctly Recognized Texts

The performance of punctuation prediction on both Chinese (Cn) and English (En) texts in the correctly recognized output of the BTEC and CT datasets are presented in Table 4 and Table 5 respectively. The

[2] http://www.cis.upenn.edu/∼treebank/tokenization.html

BTEC		No Duplication				Use Duplication				L-Crf	F-Crf
		Single Pass		Cascaded		Single Pass		Cascaded			
LM order		3	5	3	5	3	5	3	5		
Cn	*Prec.*	87.40	86.44	87.72	87.13	76.74	77.58	77.89	78.50	94.82	**94.83**
	Rec.	83.01	83.58	82.04	83.76	72.62	73.72	73.02	75.53	87.06	**87.94**
	F_1	85.15	84.99	84.79	85.41	74.63	75.60	75.37	76.99	90.78	**91.25**
En	*Prec.*	64.72	62.70	62.39	58.10	85.33	85.74	84.44	81.37	88.37	**92.76**
	Rec.	60.76	59.49	58.57	55.28	80.42	80.98	79.43	77.52	80.28	**84.73**
	F_1	62.68	61.06	60.42	56.66	82.80	83.29	81.86	79.40	84.13	**88.56**

Table 4: Punctuation prediction performance on Chinese (Cn) and English (En) texts in the correctly recognized output of the BTEC dataset. Percentage scores of precision ($Prec.$), recall ($Rec.$), and F1 measure (F_1) are reported.

CT		No Duplication				Use Duplication				L-Crf	F-Crf
		Single Pass		Cascaded		Single Pass		Cascaded			
LM order		3	5	3	5	3	5	3	5		
Cn	*Prec.*	89.14	87.83	90.97	88.04	74.63	75.42	75.37	76.87	**93.14**	92.77
	Rec.	84.71	84.16	77.78	84.08	70.69	70.84	64.62	73.60	83.45	**86.92**
	F_1	86.87	85.96	83.86	86.01	72.60	73.06	69.58	75.20	88.03	**89.75**
En	*Prec.*	73.86	73.42	67.02	65.15	75.87	77.78	74.75	74.44	83.07	**86.69**
	Rec.	68.94	68.79	62.13	61.23	70.33	72.56	69.28	69.93	76.09	**79.62**
	F_1	71.31	71.03	64.48	63.13	72.99	75.08	71.91	72.12	79.43	**83.01**

Table 5: Punctuation prediction performance on Chinese (Cn) and English (En) texts in the correctly recognized output of the CT dataset. Percentage scores of precision ($Prec.$), recall ($Rec.$), and F1 measure (F_1) are reported.

performance of the hidden event language model heavily depends on whether the duplication method is used and on the actual language under consideration. Specifically, for English, duplicating the ending punctuation symbol to the start of a sentence before training is shown to be very helpful in improving the overall prediction performance. In contrast, applying the same technique to Chinese hurts the performance.

This observed difference is reasonable and expected. An English question sentence usually starts with indicative words such as *do you* or *where* that distinguish it from a declarative sentence. Thus, duplicating the ending punctuation symbol to the start of a sentence so that it is near these indicative words helps to improve the prediction accuracy. However, Chinese presents quite different syntactic structures for question sentences. First, we found that in many cases, Chinese tends to use semantically vague auxiliary words at the end of a sentence to indicate a question. Such auxiliary words include 吗 and 呢. Thus, retaining the position of the ending punctuation symbol before training yields better performance. Another interesting finding is that, different from English, other words that indicate a question sentence in Chinese can appear at almost any position in a Chinese sentence. Examples include 哪里有... (*where* ...), ...是什么 (*what* ...), or ...多少... (*how many/much* ...). These pose difficulties for the simple hidden event language model, which only encodes simple dependencies over surrounding words by means of n-gram language modeling.

By adopting a discriminative model which exploits non-independent, overlapping features, the L-Crf model generally outperforms the hidden event language model. By introducing an additional layer of tags for performing sentence segmentation and sentence type prediction, the F-Crf model further boosts the performance over the L-Crf model. We perform statistical significance tests using bootstrap resampling (Efron et al., 1993). The improvements of F-Crf over L-Crf are statistically significant ($p < 0.01$) on Chinese and English texts in the CT

BTEC		No Duplication				Use Duplication				L-Crf	F-Crf
		Single Pass		Cascaded		Single Pass		Cascaded			
LM order		3	5	3	5	3	5	3	5		
Cn	*Prec.*	85.96	84.80	86.48	85.12	66.86	68.76	68.00	68.75	92.81	**93.82**
	Rec.	81.87	82.78	83.15	82.78	63.92	66.12	65.38	66.48	85.16	**89.01**
	F_1	83.86	83.78	84.78	83.94	65.36	67.41	66.67	67.60	88.83	**91.35**
En	*Prec.*	62.38	59.29	56.86	54.22	85.23	87.29	84.49	81.32	90.67	**93.72**
	Rec.	64.17	60.99	58.76	56.21	88.22	89.65	87.58	84.55	88.22	**92.68**
	F_1	63.27	60.13	57.79	55.20	86.70	88.45	86.00	82.90	89.43	**93.19**

Table 6: Punctuation prediction performance on Chinese (Cn) and English (En) texts in the ASR output of IWSLT08 BTEC evaluation dataset. Percentage scores of precision ($Prec.$), recall ($Rec.$), and F1 measure (F_1) are reported.

dataset, and on English texts in the BTEC dataset. The improvements of F-Crf over L-Crf on Chinese texts are smaller, probably because L-Crf is already performing quite well on Chinese. F1 measures on the CT dataset are lower than those on BTEC, mainly because the CT dataset consists of longer utterances and fewer question sentences. Overall, our proposed F-Crf model is robust and consistently works well regardless of the language and dataset it is tested on. This indicates that the approach is general and relies on minimal linguistic assumptions, and thus can be readily used on other languages and datasets.

6.2 Performance on Automatically Recognized Texts

So far we only evaluated punctuation prediction performance on transcribed texts consisting of correctly recognized words. We now present the evaluation results on texts produced by ASR systems.

For evaluation, we use the 1-best ASR outputs of spontaneous speech of the official IWSLT08 BTEC evaluation dataset, which is released as part of the IWSLT09 corpus. The dataset consists of 504 utterances in Chinese, and 498 in English. Unlike the correctly recognized texts described in Section 6.1, the ASR outputs contain substantial recognition errors (recognition accuracy is 86% for Chinese, and 80% for English (Paul, 2008)). In the dataset released by the IWSLT organizers, the correct punctuation symbols are not annotated in the ASR outputs. To conduct our experimental evaluation, we manually annotated the correct punctuation symbols on the ASR outputs.

We used all the learned models in Section 6.1, and applied them to this dataset. The evaluation results are shown in Table 6. The results show that F-Crf still gives higher performance than L-Crf and the hidden event language model, and the improvements are statistically significant ($p < 0.01$).

6.3 Performance in Translation

The evaluation process as described in Section 6.2 requires substantial manual efforts to annotate the correct punctuation symbols. In this section, we instead adopt an indirect approach to automatically evaluate the performance of punctuation prediction on ASR output texts by feeding the punctuated ASR texts to a state-of-the-art machine translation system, and evaluate the resulting translation performance. The translation performance is in turn measured by an automatic evaluation metric which correlates well with human judgments. We believe that such a task-oriented approach for evaluating the quality of punctuation prediction for ASR output texts is useful, since it tells us how well the punctuated ASR output texts from each punctuation prediction system can be used for further processing, such as in statistical machine translation.

In this paper, we use Moses (Koehn et al., 2007), a state-of-the-art phrase-based statistical machine translation toolkit, as our translation engine. We use the entire IWSLT09 BTEC training set for training the translation system. The state-of-the-art unsupervised Berkeley aligner[3] (Liang et al., 2006) is used for aligning the training bitext. We use all the default settings of Moses, except with the lexicalized reordering model enabled. This is because

[3] http://code.google.com/p/berkeleyaligner/

	No Duplication				Use Duplication				L-CRF	F-CRF
	Single Pass		Cascaded		Single Pass		Cascaded			
LM order	3	5	3	5	3	5	3	5		
Cn → En	30.77	30.71	30.98	30.64	30.16	30.26	30.33	30.42	31.27	**31.30**
En → Cn	21.21	21.00	21.16	20.76	23.03	24.04	23.61	23.34	23.44	**24.18**

Table 7: Translation performance on punctuated ASR outputs using Moses (Averaged percentage scores of BLEU)

lexicalized reordering gives better performance than simple distance-based reordering (Koehn et al., 2005). Specifically, the default lexicalized reordering model (*msd-bidirectional-fe*) is used.

For tuning the parameters of Moses, we use the official IWSLT05 evaluation set where the correct punctuation symbols are present. Evaluations are performed on the ASR outputs of the IWSLT08 BTEC evaluation dataset, with punctuation symbols inserted by each punctuation prediction method. The tuning set and evaluation set include 7 reference translations. Following a common practice in statistical machine translation, we report BLEU-4 scores (Papineni et al., 2002), which were shown to have good correlation with human judgments, with the closest reference length as the effective reference length. The minimum error rate training (MERT) (Och, 2003) procedure is used for tuning the model parameters of the translation system. Due to the unstable nature of MERT, we perform 10 runs for each translation task, with a different random initialization of parameters in each run, and report the BLEU-4 scores averaged over 10 runs.

The results are reported in Table 7. The best translation performances for both translation directions are achieved by applying F-CRF as the punctuation prediction model to the ASR texts. Such improvements are observed to be consistent over different runs. The improvement of F-CRF over L-CRF in translation quality is statistically significant ($p < 0.05$) when translating from English to Chinese. In addition, we also assess the translation performance when the manually annotated punctuation symbols as mentioned in Section 6.2 are used for translation. The averaged BLEU scores for the two translation tasks are 31.58 (Chinese to English) and 24.16 (English to Chinese) respectively, which show that our punctuation prediction method gives competitive performance for spoken language translation.

It is important to note that in this work, we only focus on optimizing the punctuation prediction performance in the form of F1-measure, without regard to the subsequent NLP tasks. How to perform punctuation prediction so as to optimize translation performance is an important research topic that is beyond the scope of this paper and needs further investigation in future work.

7 Conclusion

In this paper, we have proposed a novel approach for predicting punctuation symbols for transcribed conversational speech texts. Our proposed approach is built on top of a dynamic conditional random fields framework, which jointly performs punctuation prediction together with sentence boundary and sentence type prediction on speech utterances. Unlike most previous work, it tackles the task from a purely text processing perspective and does not rely on prosodic cues.

Experimental results have shown that our proposed approach outperforms the widely used approach based on the hidden event language model, and also outperforms a method based on linear-chain conditional random fields. Our proposed approach has been shown to be general, working well on both Chinese and English, and on both correctly recognized and automatically recognized texts. Our proposed approach also results in better translation accuracy when the punctuated automatically recognized texts are used in subsequent translation.

Acknowledgments

This research was done for CSIDM Project No. CSIDM-200804 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

References

D. Beeferman, A. Berger, and J. Lafferty. 1998. CYBERPUNC: A lightweight punctuation annotation system for speech. In *Proc. of ICASSP'98*.

S.F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL'06*.

H. Christensen, Y. Gotoh, and S. Renals. 2001. Punctuation annotation using statistical prosody models. In *Proc. of ISCA Workshop on Prosody in Speech Recognition and Understanding*.

B. Efron, R. Tibshirani, and R.J. Tibshirani. 1993. *An introduction to the bootstrap*. Chapman & Hall/CRC.

B. Favre, R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tur, and M. Ostendorf. 2008. Punctuating speech for information extraction. In *Proc. of ICASSP'08*.

A. Gravano, M. Jansche, and M. Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *Proc. of ICASSP'09*.

D. Hillard, Z. Huang, H. Ji, R. Grishman, D. Hakkani-Tur, M. Harper, M. Ostendorf, and W. Wang. 2006. Impact of automatic comma prediction on POS/name tagging of speech. In *Proc. of SLT'06*.

J. Huang and G. Zweig. 2002. Maximum entropy model for punctuation annotation from speech. In *Proc. of ICSLP'02*.

J.H. Kim and P.C. Woodland. 2001. The use of prosody in a combined system for punctuation generation and speech recognition. In *Proc. of EuroSpeech'01*.

K. Kirchhoff and M. Yang. 2007. The University of Washington machine translation system for the IWSLT 2007 competition. In *Proc. of IWSLT'07*.

P. Koehn, A. Axelrod, A.B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. of IWSLT'05*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL'07 (Demo Session)*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML'01*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Proc. of HLT/NAACL'06*.

Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *Proc. of ACL'05*.

E. Matusov, A. Mauser, and H. Ney. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proc. of IWSLT'06*.

A. McCallum, K. Rohanimanesh, and C. Sutton. 2003. Dynamic conditional random fields for jointly labeling multiple sequences. In *Proc. of NIPS'03 Workshop on Syntax, Semantics and Statistics*.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL'03*.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL'02*.

M. Paul. 2008. Overview of the IWSLT 2008 evaluation campaign. In *Proc. of IWSLT'08*.

M. Paul. 2009. Overview of the IWSLT 2009 evaluation campaign. In *Proc. of IWSLT'09*.

S. Sarawagi and W.W. Cohen. 2005. Semi-Markov conditional random fields for information extraction. In *Proc. of NIPS'05*.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. of HLT-NAACL'03*.

A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu. 1998. Automatic detection of sentence boundaries and disfluencies based on recognized words. In *Proc. of ICSLP'98*.

A. Stolcke. 2002. SRILM–an extensible language modeling toolkit. In *Proc. of ICSLP'02*.

C. Sutton and A. McCallum. 2004. Collective segmentation and labeling of distant entities in information extraction. In *Proc. of ICML'04 workshop on Statistical Relational Learning*.

C. Sutton, A. McCallum, and K. Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8.

C. Sutton. 2006. GRMM: GRaphical Models in Mallet. `http://mallet.cs.umass.edu/grmm/`.

H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing*.

M. Wainwright, T. Jaakkola, and A. Willsky. 2001. Tree-based reparameterization for approximate inference on loopy graphs. In *Proc. of NIPS'01*.

H. Wang, H. Wu, X. Hu, Z. Liu, J. Li, D. Ren, and Z. Niu. 2008. The TCH machine translation system for IWSLT 2008. In *Proc. of IWSLT'08*.