MAXIMUM ENTROPY MODEL FOR PUNCTUATION ANNOTATION FROM SPEECH

Jing Huang and Geoffrey Zweig

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
USA
jhuang, gzweig@watson.ibm.com

ABSTRACT

In this paper we develop a maximum-entropy based method for annotating spontaneous conversational speech with punctuation. The goal of this task is to make automatic transcriptions more readable by humans, and to render them into a form that is useful for subsequent natural language processing and discourse analysis. Our basic approach is to view the insertion of punctuation as a form of tagging, in which words are tagged with appropriate punctuation, and to apply a maximum entropy tagger that uses both lexical and prosodic features. We present experimental results on Switchboard data with both reference transcriptions and transcriptions produced by a speech recognition system.

1. INTRODUCTION

With the advance of automatic speech recognition technology, large amounts of audio data such as meeting and call-center recordings can now be transcribed automatically. While this results in dramatic labor savings, the automatically generated transcripts are just sequences of words with no sentence boundaries, no punctuation and no casing. Therefore, it is hard for humans to read and understand them, and valuable clues that might be used in natural language understanding or information retrieval are not available.

The problem of sentence punctuation arises in a number of different contexts, but this paper we focus on spontaneous conversational speech data in the form of the Switchboard corpus. One of the main characteristics of this kind of data is the presence of speech disfluencies, such as filled pauses, false starts and repairs. These phenomena cause significant problems for ASR algorithms, and also decrease the readability of the output transcripts. Improving readability by inserting punctuation is thus especially important in this domain.

A number of recent papers [1, 2, 3] have examined the punctuation problem, and suggest that textual, acoustic, and prosodic features can all be used in determining appropriate punctuation. A couple of examples illustrate the issues.

A: Um, we moved from Colorado where...
B: He lives in Virginia, now?

In the first sentence, the presence of the filler word "Um" gives a strong indication that a comma should follow. In the second sentence, however, prosodic information is the key to determine whether the punctuation at the end of the sentence should be marked as a question mark or a period. Further evidence of the potential importance of prosodic information is given in [4] which shows that information such as pause duration and pitch change is highly correlated with the positions of punctuation marks and discourse structure of conversations.

The maximum entropy modeling provides a easy and natural framework to incorporate both textual and prosodic information in punctuation annotation. Here we treat the punctuation as a tagging problem: each word is tagged with one of several possible punctuation marks: comma, period and question mark, or a default (denoted by X). By defining features that use combinations of textual and prosodic features, we are able to create integrated models that use all the available forms of information.

The rest of the paper is organized as follows: Section 2 reviews recent work on punctuation annotation; Section 3 describes the maximum entropy model and the associated features; Section 4 contains our experimental setup and results and Section 5 concludes our discussions.

2. RELATED WORK

In recent years, several approaches to punctuating the output of a speech recognizer have been investigated. One of the first attempts was Cyberpunc: a light-weight punctuation annotation system for speech [5]. This system exploited textual features, and focused on identifying commas. Cyberpunc worked by building an extended trigram language model containing punctuation marks as words, and finding the set of comma insertions (possibly none) that would maximize the resulting language model probability.

When punctuation marks are generated simultaneously

with the recognized words, they can be treated as word entities which have associated acoustic pronunciations [1]. By using the acoustic baseforms of silence, breath, and other non-speech sounds to represent punctuation marks, and using a language model that was built with punctuation, [1] reported that the system was able to identify punctuation such as commas, periods, colons, and question marks.

Adopting the combination strategy of [6], Kim and Woodland [3] model prosodic features with a CART-style decision tree, and then combine these probabilities with those generated by a language model. Experiments are conducted on the Broadcast News corpus and evaluated on the identification of commas, periods and question marks. It is reported that F-measure performance significantly increased when prosodic features were used to punctuate the output of a speech recognizer. Further, when the reference scripts were used, the prosodic model alone was better than the language model, with further improvements from combination.

Christensen et al. also investigated punctuation annotation for Broadcast News data [2], again focusing on periods, commas, and question marks. That work investigates the use of both finite state and neural-net based methods, and suggests that both are reasonable approaches. The MLP experiments further show that pause duration features are the strongest candidates for automatic punctuation; other prosodic features including vowel duration, phone duration, pitch slope, pitch range and pitch mean did not help. The experiments are done on the BN reference transcripts, and the results can not be compared to those in [3] because (1) the test data are different; (2) the performance measures are different. In [3], a half score is given when a punctuation is located right but recognized as a wrong type of punctuation. Thus the scores of precision and recall are made higher than those from the usual definitions of precision/recall in [2].

3. MAXIMUM ENTROPY MODEL

Maximum entropy (Maxent) modeling is a powerful framework for constructing statistical models from data. It has been used in a variety of difficult classification tasks such as part-of-speech tagging [7], prepositional phrase attachment [8] and named entity tagging [9], and achieves state of the art performance. We have also applied the maxent model to extracting caller-information from voicemail messages [10], with good results.

3.1. Tagging

The problem of annotating words with punctuation marks can also be thought of as a tagging problem [10], where the possible tags are comma (,), period (.), question mark (?) and the default tag (X). The objective is to tag each word in a message with one of these categories. The information that can be used to predict the tag of a word includes context of its surrounding words, their associated tags, and prosodic features of those words.

Let $\mathcal{H}$ denote the set of possible combined contexts, called "*histories*", and $\mathcal{T}$ denote the set of tags. The maxent model is then defined over $\mathcal{H} \times \mathcal{T}$,and predicts the conditional probability $p(t|h)$ for a tag t given the history h. The computation of this probability depends on a set of binary-valued "features" $f_i(h, t)$ as follows:

$$p(t|h) = \frac{\prod_i \alpha_i^{f_i(h,t)}}{Z}$$

where Z is a normalization constant.

The role of the features is to enumerate co-occurrences of histories and tags, and to find histories that are strong predictors of specific tags. (for example, the tag "," often goes with filler words "um" and "yeah"). If a feature is a very strong predictor of a particular tag, then the corresponding α_i would be high. It is also possible that a particular feature may be a strong predictor of the absence of a particular tag, in which case the associated α_i would be near zero.

Training a maximum entropy model involves the selection of the features and the subsequent estimation of weight parameters α_i. The testing procedure involves a search to enumerate the candidate tag sequences for a message and choosing the one with highest probability. We use the "beam search" technique of [7] to search the space of all hypotheses.

3.2. Features

Designing effective features is crucial to maximum entropy modeling. In the following, we briefly describe how we design the lexical and prosodic features, and how to combine them in the maxent framework.

We first experimented with lexical features. We used the neighboring two words, and the tags associated with the previous two words to define the history h_i as

$$h_i = w_i, w_{i+1}, w_{i+2}, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}$$

The features are generated by scanning each pair (h_i, t_i) in the training data with unigram and bigram feature templates as in [10]. Figure 1 illustrates the unigram feature template.

In the light of the results of [2], we decided to use pause duration features to incorporate prosodic information into our models. Pause duration is measured in 0.01-second intervals. Again, unigram and bigram pause features were created from templates. To combine both linguistic and prosodic information in an maxent model, we just put the lexical history and pause history into one, as illustrated in Figure 1:

	Features		
$\forall w_i$	$w_i = X$	&	$t_i = T$
	$t_{i-1} = X$	&	$t_i = T$
	$t_{i-2}t_{i-1} = XY$	&	$t_i = T$
	$w_{i-1} = X$	&	$t_i = T$
	$w_{i-2} = X$	&	$t_i = T$
	$w_{i+1} = X$	&	$t_i = T$
	$w_{i+2} = X$	&	$t_i = T$

Table 1. Unigram features of the current history h_i.

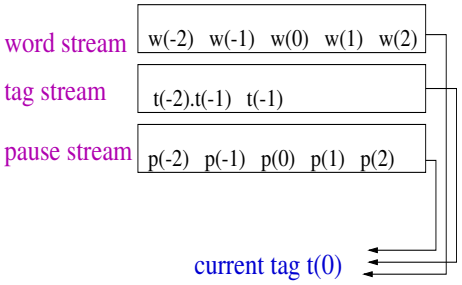

Fig. 1. Adding a code stream to the history h_i. Features can refer to all three streams.

4. EXPERIMENTAL RESULTS

We use the Switchboard corpus released by LDC, which is punctuated with commas, periods and question marks. There are total of $210,000$ lines of conversation; we used 90% of them used for training data, and 10% for cross-validation (CV) data. The test data is the Hub5-2000 evaluation data, with 1831 lines of conversation. We present results for both reference transcripts, and for the decoded transcripts of a speech recognizer. The WER of Hub5 eval'00 is about 20%.

We use the conventional *precision*, *recall*, and their *F-measure* as used in [2]. to evaluate performance. Using C, I, M, and S to denote the numbers of correct, incorrect, missing, and spurious identifications, recall is defined as $R = C/(C + I + M)$; precision is $P = C/(C + I + S)$; and $F = 2PR/(P+R)$. The higher P/R/F scores, the better performance.

Tables 2, 3, and 4 present results on CV data when reference transcripts are used. In these tables, the MEl/p-U model uses unigram lexical/pause features only, and the MEl/p-B model uses bigram lexical/pause features only. From Table 2, we see that compared to MEl-U, MEl-B improves recall by 6% absolute, but only marginally by 1% on the precision. Overall, bigrams improve the F-measure by 4% absolute over unigrams, with about double number of features.

Moving to Table 3, we observe that contrary to the results reported in [3] our lexical-based maxent model (with an F-measure of 0.79) is much better than the pause-based maxent model (with an F-measure of 0.38). The reason may be that fact that the style of conversational speech (Switchboard) is quite different from that of the read speech (Broadcast News), or the vocabulary of the maxent model for pause duration is too small (about $3k$ compared to $30k$ for the lexical maxent model). This issue needs to be resolve in the future.

	P(U)	R(U)	F(U)	P(B)	R(B)	F(B)
Period	0.80	0.88	0.84	0.80	0.84	0.82
Comma	0.85	0.62	0.72	0.87	0.73	0.79
Ques. mark	0.57	0.28	0.38	0.64	0.28	0.39
Overall	**0.83**	**0.69**	**0.75**	**0.84**	**0.75**	**0.79**

Table 2. Precision and recall rates using lexical features only (the MEl-U and MEl-B models) on the CV data.

	P(U)	R(U)	F(U)	P(B)	R(B)	F(B)
Period	0.64	0.70	0.67	0.64	0.70	0.67
Comma	0.52	0.07	0.12	0.51	0.10	0.17
Ques. mark	0	0	-	0	0	-
Overall	**0.61**	**0.26**	**0.36**	**0.60**	**0.28**	**0.38**

Table 3. Precision and recall rates using prosodic features only (the MEp-U and MEp-B models) on the CV data.

Table 4 presents results using both lexical and pause features. Compared to the MEl-U model or the MEl-B model, adding pause features only improves the recall by 1% respectively. This is contrary to the results reported in [3] and [2], where adding the prosody model improves on the language model. There are several possible reasons: (1) the databases are different, one is Hub5, one is Hub4; (2) our lexical maxent models alone perform very well; (3) the prosodic quantization scheme is suboptimal. To roughly compare to the results reported in [2], we trained a lexical unigram maxent model on the Hub4 Broadcast News data, with 100 shows of training data and 14 shows of test data - roughly the same amount as [2]. Our results are: $P = 0.55, R = 0.33, F = 0.41$. This compares with those in [2] that additionally use prosodic features: $p = 0.46, R = 0.17, F = 0.25$. Although the test sets are different, this does indicate that our absolute numbers are reasonable.

After examining the results more carefully, there are several observations that can be made. Commas are often confused with the default X, and question marks are often confused with periods. Lexically, this is likely because the features that distinguish these marks can a span that is much longer than bigrams. Further, little prosodic information is available because commas and X both have very short typical pause durations, and both periods and question marks usually have long pause durations.

	P(U)	R(U)	F(U)	P(B)	R(B)	F(B)
Period	0.80	0.89	0.84	0.79	0.85	0.82
Comma	0.85	0.63	0.73	0.87	0.73	0.79
Ques. mark	0.56	0.29	0.38	0.65	0.27	0.38
Overall	**0.83**	**0.70**	**0.76**	**0.84**	**0.76**	**0.80**

Table 4. Precision and recall rates using both lexical and prosodic features (the MElp-U and MElp-B models) on the CV data.

	P(U)	R(U)	F(U)	P(B)	R(B)	F(B)
Period	0.73	0.65	0.69	0.74	0.65	0.69
Comma	0.78	0.61	0.69	0.77	0.74	0.76
Ques. mark	0.45	0.17	0.25	0.64	0.14	0.23
Overall	**0.76**	**0.61**	**0.68**	**0.76**	**0.70**	**0.73**

Table 5. Precision and recall rates for different punctuation marks for the MElp-U model and the MElp-B model on the eval'00 test data.

Table 5 presents results of punctuation on Hub5 eval'00 data with speech recognition WER of 20%. The decoded scripts were manually punctuated and served as reference punctuation scripts, rather than using the punctuation from the reference scripts as the correct punctuation. This is because our goal is to annotate the decoded scripts to make them more readable and for further natural language processing. Overall, the results on the test data degrade about 7% absolute with respect to those for the CV data, where the training and testing data are both drawn from reference scripts. The F-measure is still reasonably good at 70%.

5. CONCLUSION AND DISCUSSION

In this paper we develop a maximum-entropy based approach for annotating spontaneous conversational speech with punctuation marks. The goal of this task is to make transcriptions from an automatic speech recognizer more readable, and to make these transcripts useful for subsequent natural language processing and discourse analysis. Our approach is to view the insertion of punctuation as a form of tagging, in which words are tagged with appropriate punctuation, and to apply a maximum entropy tagger that uses both lexical and prosodic features. Our experimental results on Switchboard data with reference transcriptions achieve 80% in F-measure, and 73% F-measure for transcriptions produced by a speech recognition system.

In future work, there are several issues we would like to explore. The first of these is to study other kinds of prosodic features. The second issue involves the evaluation metric. Although typical uses of punctuation are documented in standard reference books, the style and functions of punctuation marks vary from person to person, and from domain to domain. Therefore, as was pointed out in [1], the absolute accuracy of punctuation of a given text may not be the optimal measure of success. Here, user-studies focusing on task-completion times may provide guidance in defining better evaluation metrics.

6. REFERENCES

[1] C. Julian Chen. 1999. Speech Recognition with Automatic Punctuation. In *Proceedings of Eurospeech*, pages 447–450, 1999, Budapest, Hungary.

[2] Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. 2001. Punctuation Annotation using Statistical Prosody Model. In *Proceedings of Eurospeech* 2001, Aalborg, Denmark.

[3] Ji-Hwan Kim and P. C. Woodland. 2001. The Use of Prosody in a Combined System for Punctuation Generation and Speech Recognition. In *Proceedings of Eurospeech* 2001, Aalborg, Denmark.

[4] E. Shriberg et al. 1998. Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech? *Language and Speech*, 41:439–487, 1998.

[5] D. Beeferman, A. Berger, and J. Lafferty. 1998. A Lightweight Punctuation Annotation system for Speech. In *Proc. ICASSP*, pages 689–692, 1998.

[6] D. hakkani-Tur, G. Tur, A. Stolcke, and E. Shriberg. 1999. Combining Words and Prosody for Information Extraction from Speech. In *Proceedings of Eurospeech*, pages 1991–1994, 1999, Budapest, Hungary.

[7] Adwait Ratnaparkhi. 1996. A Maximum Entropy Part of Speech Tagger. In Eric Brill and Kenneth Church, editors, *Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania, May 17–18.

[8] Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the Human Language Technology Workshop*, pages 250–255, Plainsboro, N.J. ARPA.

[9] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. Nyu: Description of the mene named entity system as used in MUC-7. In *Seventh Message Understanding Conference(MUC-7)*. ARPA.

[10] Geoffrey Zweig, Jing Huang and Mukund Padmanabhan. 2001. Extracting Caller Information from Voicemail. In *Proceedings of Eurospeech* 2001, Aalborg, Denmark.