IMPACT OF AUTOMATIC COMMA PREDICTION ON POS/NAME TAGGING OF SPEECH

D. Hillard†*, Z. Huang*‡*, H. Ji*, R. Grishman*, D. Hakkani-Tur**, M. Harper*‡*, M. Ostendorf*†*, W. Wang*††

†University of Washington, Electrical Engineering Dept., Seattle, WA
‡Purdue University, School of Electrical and Computer Engineering, West Lafayette, IN
* New York University, Computer Science Dept., New York, NY
** ICSI, Berkeley, CA USA ††SRI International, Menlo Park, CA

ABSTRACT

This work looks at the impact of automatically predicted commas on part-of-speech (POS) and name tagging of speech recognition transcripts of Mandarin broadcast news. There is a significant gain in both POS and name tagging accuracy due to using automatically predicted commas over sentence boundary prediction alone. One difference between Mandarin and English is that there are two types of commas, and experiments here show that, while they can be reliably distinguished in automatic prediction, the distinction does not give a clear benefit for POS or name tagging.

Index Terms— natural language, speech recognition

1. INTRODUCTION

As large vocabulary automatic speech recognition (ASR) technology has dramatically improved in the past few years, it is now possible to explore language processing on speech sources as well as text. Information extraction and summarization of broadcast news is of particular interest, because of the large number of such information sources available and the generally higher recognition accuracy for news sources (when compared to other domains).

One of the key differences between speech and text sources, other than the potential for transcription errors, is that typical ASR systems do not output punctuation cues, which are used in most text processing systems. In [1], researchers from BBN showed that missing commas can have a dramatic impact on information extraction performance compared to using hand-transcribed commas, with performance losses typically bigger than that for moving from reference to automatic sentence segmentation (for a range of word error rates on English news). In the current work, we confirm these results for Mandarin and further look at how much performance can be recovered using automatically predicted commas. We examine name tagging, as in the BBN study, but also look at part-of-speech tagging which benefits name tagging (and other NLP tasks) as a pre-processing step. In addition, since the role of the English comma is split in Chinese to distinguish commas separating words in a list (caesura) from other uses, we consider the question of whether the distinction between the two different comma types in Mandarin can be reliably predicted and whether it is useful for the two tagging tasks.

The overall system architecture used here involves running automatic speech recognition, then punctuation prediction, then part-of-speech tagging, and finally name tagging. In the next sections, we describe the baseline components, the corpora and evaluation paradigm used in the studies, experimental results on punctuation prediction and its impact on the tagging tasks, and finally conclude with a summary of the key findings.

2. COMPONENT SYSTEMS

The speech recognition system used in this work is a state-of-the-art system, based on the SRI Decipher recognizer [2] and trained/tuned specifically for the Mandarin broadcast news task. Training texts for the system from a variety of sources were automatically word-segmented, using a maximum n-gram probability criterion as in [3] and using all punctuation marks as delimiters during segmentation. The top 60k words were used as the decoding vocabulary, which includes several thousand frequent Chinese person names. The recognizer combines cepstra, pitch, and neural network phone posteriors as features, and uses MPE training, cross-system adaptation, and a 5-gram mixture language model with components from 9 separate text sources. On the broadcast news development set used in these experiments, character error rate is 5.6%.

We utilized two part-of-speech taggers. The first is a Viterbi tagger that builds on the tagger developed in [4] that uses trigram transition probability $P(T_i|T_{i-1}T_{i-2})$ and trigram emission probability $P(W_i|T_iT_{i-1})$, where T_i and W_i represent the i-th tag and word. When a word was not observed during training (unknown word), it estimates the emission probability as a weighted sum of $P(S_i^k|T_iT_{i-1})$, where S_i^k is the k-th suffix in word W_i. When applied to the LDC Chinese Treebank 5.2, the tagger obtained a tagging accuracy of 93.6% (69.2% on unknown words). However, the accuracy of the tagger was improved to 94.5% (76.8% on unknown words) by enriching the context model in two ways: the emission probability is replaced by $P(W_i|T_iT_{i-1})^{\frac{1}{2}} \times P(W_{i-2}|T_{i-2}T_{i-1})^{\frac{1}{2}}$ for both known and unknown words, and $P(W_i|T_iT_{i-1})$ is replaced by the geometric mean of $P(C_i^k|T_iT_{i-1})$ for all the characters C_i^k in any unknown word W_i (and similarly for $P(W_{i-2}|T_{i-2}T_{i-1})$). The second POS tagger, which utilizes the N-best extraction of the first tagger, incorporates various higher order N-gram features in a reranking method based on the boosting approach described in [5]. This tagger improves

M. Harper is also with CASL, University of Maryland, College Park, MD. This work was supported by DARPA grant MDA972-02-C-0038. Any opinions, findings and/or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

tagging accuracy to 94.84% (76.98% on unknown words).

The name tagger is based on an HMM that generally follows the Nymble model [6]. It identifies names of three classes: people, organizations, and locations. Nymble used an HMM with a single state for each name class, plus one state for non-name tokens. To take advantage of the structure of Chinese names, we used a model with a larger number of states, 14 in total. The expanded HMM can handle name prefixes and suffixes, and has separate states for transliterated foreign names. The HMM was supplemented with a set of post-processing rules to correct some omissions and systematic errors. Some of these rules are dependent on the part-of-speech tags assigned to the tokens.

3. CORPORA AND EVALUATION

Different corpora were used for training the various component systems, as described in the respective sections. In all cases, text normalization was needed to get rid of phrases with bad or corrupted codes, and convert numbers, dates and currencies into their verbalized forms in Chinese. Among these, number normalizations were performed using a set of context-independent and context-dependent heuristic rules. Then automatic word segmentation was run, using all punctuation marks as delimiters. For training most systems, we kept sentence boundary punctuation marks, comma and caesura marks, and removed all other punctuation marks.

The speech test set in this work includes transcripts from the GALE Mandarin ASR/MT development test set, where we use the four dev shows from the GALE Year 1 BN audio release. The data set includes about 15k words (about 26k characters). To avoid over-tuning on this set, all text data from months covered by these shows are excluded from training.

The target data for this work is automatically transcribed speech, specifically Mandarin broadcast news, but there is no such speech data with hand-annotated part-of-speech tags and name labels. For that reason, most of the development work involved text corpora, where annotated data is available and precision/recall can easily be measured. For experiments with speech, we have adopted a change comparison method to assess the impact of comma prediction on both POS and name tagging accuracy for speech recognition output. Specifically, human annotators examine only those tokens for which the automatic POS (or name) predictions differ on the speech recognition output and assess whether the change corrects or introduces an error, with access to the reference transcription.

4. PUNCTUATION PREDICTION

Previous work on punctuation detection to enhance the output of automatic speech recognizers mostly focuses on sentence segmentation [7, 8, 9]. For intra-sentence punctuation insertion for text, Beeferman *et al.* [10] use lexical information in the form of trigram language models. Zhang *et. al.* [11] use decision trees with linguistically sophisticated features for enriching natural language generation output, and obtain better results than using n-gram language models.

In this work, we use the ICSI+ multi-lingual sentence segmentation tools [12] for both comma and sentence boundary detection. The sentence boundary detection is treated as a classification problem, where every word boundary can be of one of two classes: sentence boundary vs. non-sentence boundary. The classifier uses a combination of hidden-event language models (5-gram) to exploit lexical information and sequence dependencies, and a boostexter classifier [13] to exploit lexical cues (word triples) in combination with prosodic and speaker change information. Prosodic features include various measures and normalizations of pause duration, phone duration, fundamental frequency and energy. The posteriors from the two models are interpolated using equal weights. The SRI-LM toolkit [14] (with Kneser-Ney smoothing) is used for training the hidden-event model for sentence boundary prediction, with the same data sources as for training the Mandarin ASR language models, including broadcast news speech transcripts, TDT text data, the Chinese Gigaword corpus, the Chinese portion of various news translation corpora, and web news data collections from National Taiwan University and Cambridge University.

The approach to comma prediction is similar to sentence boundary prediction. The same features apply, with the exception of speaker change (because it is not informative for sentence internal events), and prosodic features are only included for the speech experiments. While comma and sentence boundary prediction could be treated jointly as a multi-class problem, in this work we take predicted sentence boundaries as given and then predict commas within the sentence in order to factor out the impact of comma prediction on sentence prediction from the impact of commas per se.

The comma hidden-event language model is trained on the Chinese Gigaword corpus, where the training text has been stripped of all punctuation but comma and caesura. (Preliminary experiments showed that interpolation with other sources did not give a gain, and some other sources did not distinguish between commas and caesura.) The boostexter model is trained on a subset of the TDT4 Chinese news data (40 shows) using flexible alignment [15] to obtain word transcripts from the closed-captions. For performance analysis, reference punctuation is determined by first aligning the ASR words to reference words, and then choosing the best punctuation alignment if multiple word alignments are equivalent.

Figure 1 shows the precision/recall curve for commas (using ASR words and reference sentence boundaries) on a held-out set of 10 shows from our TDT4 set. The best results are obtained with the combined model, but the individual component models both give reasonable performance alone. We have merged the two commas into one class for this figure to evaluate only comma position performance, ignoring comma type. We found a slight improvement in the merged comma prediction by modeling the two types of commas separately and mapping them to a single comma afterwards (rather than training on merged commas).

Table 1 shows confusion between comma and caesura types on the TDT4 heldout set. There are many fewer caesuras, but even with the highly skewed distribution there is very little confusion between the two comma types.

The speech test set we use for POS and name tagging includes about 1,400 commas (with no comma/caesura distinction) and 600 sentences. Results for comma prediction on this data are given in Table 2 for automatically detected sentence boundaries (using a threshold of 0.5 for the sentence posterior). For this condition, senten ce boundary detection performance is P=0.53, R=0.79, and F=0.63, so sentence boundary recall is much higher than precision. While precision for the commas with automatic sentence boundaries is sim-

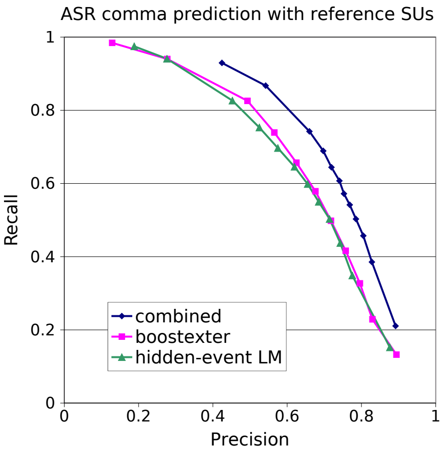

Fig. 1. Comma prediction for TDT4 ASR words with reference sentence boundaries and three different modeling approaches.

Table 1. Confusion table counts for comma and caesura prediction on the TDT held out set, using a .5 comma threshold.

	Predicted		
True	comma	caesura	null
comma	2,924	8	2,049
caesura	32	104	245
null	992	12	74,207

ilar to using reference sentences, recall is significantly lower. The primary reason is that many "false" sentence boundaries are hypothesized at comma locations, limiting the possible recall. When selecting sentence boundaries with the $.5$ threshold used here, 27% of the reference comma locations are unavailable due to being marked as sentence boundaries, while with a $.8$ sentence boundary threshold, 12% are unavailable.

5. PART-OF-SPEECH TAGGING

In an attempt to optimize our tagger for the condition of tagging speech transcripts with automatically generated commas and caesuras, we performed a series of experiments on textual data to determine the impact of punctuation on tagging accuracy, and to assess the best conditions to train our tagger when using automatically generated commas. We also report the no punctuation (lower bound) and all punctuation (upper bound) performance levels. For these experiments, before scoring the tag sequence, we remove all punctuation along with their tags to more fairly compare the tag accuracy on words resulting from the absence or presence of punctuation of various qualities. These studies, selectively reported in Table 3, used the LDC Chinese Treebank 5.2 with 10-fold cross-validation. Because the Viterbi tagger is computationally efficient, we utilized it in all the experimental conditions, but report selective results with the reranking tagger to highlight the best possible performance with our current systems.

The best tagging results overall were obtained when training and testing on matched conditions. For example, if we train the tagger using all of the Treebank punctuation and then apply it to tag word sequences with automatically generated commas, there is a serious degradation in tagging accuracy (e.g., 92.90% using comma and caesura predictions). We also found that using a comma prediction threshold of 0.5 (out of 0.2, 0.5, and 0.8) gave the best accuracy. There is a negligible improvement from keeping the distinction between comma and caesura, rather than merging the two to a single comma type. The best overall tagging accuracy for comma predicted data was obtained when using a comma threshold of 0.5 after training with word sequences annotated with gold commas, no commas, and predicted commas.

Table 2. Results for comma detection on ASR transcripts with different thresholds for comma posteriors. Automatic sentence boundary detection F=0.63.

Thresh	Rec	Prec	F
0.2	0.26	0.67	0.38
0.5	0.20	0.69	0.31
0.8	0.15	0.73	0.25

We evaluated the impact of automatic comma prediction on POS tagging accuracy on the ASR output for the speech test set. We compared tagging results using our Viterbi tagger under two conditions. For the first case without punctuation, the ASR output was tagged by a tagger trained on the Treebank with all punctuation removed. For the second case, in which ASR output was augmented with predicted commas and caesuras with a 0.5 threshold, the best setup in Table 3 was used. Three annotators were asked to compare the POS tag changes in the two tagging outputs, without knowing in advance which system they came from. To support the comparison, we adapted an emacs tagging tool used by LDC to highlight the differences and mark up whether the change was from incorrect to correct, correct to incorrect, or incorrect to some other incorrect tag. All tag changes related to word segmentation and/or ASR errors were discarded. If the POS of a word could not be agreed upon among the annotators, then the majority vote was used for scoring (or the tag change of the word was discarded when all annotators disagreed). Of the 247 differences between the no punctuation and the automatic comma prediction tagging outputs, 29 were discarded, 120 were positive changes, 69 were negative changes, and 29 were wrong in both cases. Hence, the predicted commas significantly improved POS tagging accuracy ($p <= 0.00027$ using the sign test).

6. NAME TAGGING

The name tagger was trained on 585 documents from the training data for the 2005 ACE (Automatic Content Extraction) evaluation, containing 198k words. Three separate name tagger models were trained: one with all sentence-internal punctuation in the training texts removed, one with the commas and caesuras added automatically, and one with the reference commas and caesuras retained.

The effect of comma prediction on named entity tagging was then evaluated using two test corpora, a text corpus and an ASR transcript. The text corpus consisted of 50 documents from the ACE 2005 training set (42 manually-prepared broadcast news transcripts and 8 newswire articles), containing a total of 2,671 names. The results are shown in Table 4. The first row represents a system trained and tested without commas; the last row a system trained and tested with the com-

Table 3. POS tagging performance on various training/test conditions using the Viterbi [and reranking] tagger. The last group of results was obtained by joining the three training sets (without punctuation, with Treebank commas, and with predicted comas) into one large training set.

Punctuation Source	**Training Punctuation**	**Testing Punctuation**	**Accuracy (%) Viterbi[Reranked]**
None	None	None	92.99 [93.28]
Treebank	All	All	93.49 [93.91]
	Merged comma	Merged comma	93.40
	Caesura/comma	Caesura/comma	93.44
Prediction	Merged comma	Merged comma	93.13
	Caesura/comma	Caesura/comma	93.14
Combination	Merged comma	Merged comma	93.16
	Caesura/comma	Caesura/comma	93.17 [93.46]

Table 4. Named entity tagging performance on news text under different punctuation conditions.

	Recall	Precision	F-measure
No commas	85.1	84.7	84.9
Comma prediction	85.6	85.1	85.4
True commas	85.7	86.1	85.9

mas and caesuras from the original corpora. True commas produced a 1% gain in NE F-measure. The intervening row shows the results using comma prediction (with the system distinguishing commas and caesuras); this yields half the gain (0.5% in F-measure) of the true-comma case. The comma predictions changed the tagging of 36 tokens in the text test corpus: 26 incorrect tags were corrected, 9 correct tags were changed to incorrect ones, and 1 incorrect tag was changed to other incorrect tag (significant at $p <= .006$ by the sign test). The performance when not distinguishing commas and caesuras was slightly but not significantly worse – 0.1% lower in F-measure.

The ASR test corpus, as for the POS tests, was the speech test set drawn from GALE Y1 Mandarin ASR+MT common dev set, and included 881 sentences with approximately 1700 names. The comma predictions changed the tagging of 59 tokens in the test corpus; 44 incorrect tags were corrected, 9 correct tags were changed to incorrect ones, and 6 incorrect tags were changed to other incorrect tags. The predicted commas significantly improved name tagging accuracy ($p <= 0.000002$ using the sign test). Two native speakers independently evaluated the changes and then adjudicated their decisions; the independent assessments agreed 94% of the time.

In examining the changes, we observed a number of cases where the comma predictor was able to predict a comma before a name, and this enabled the name tagger to identify a name that it had previously missed, or to correct a name boundary error. For example, in the sentence (translating the actual Chinese example):

> More than 200 pictures including the masterpieces by [Zhang Daqian]$_{PER1}$, [Zhao Zhi Qian]$_{PER2}$, [Xu Beihong]$_{PER3}$ and [Qi Baishi]$_{PER4}$ etc. were on sale in [Shanghai]$_{LOC}$.

The second name, "Zhao Zhi Qian" is missed when commas are not present because the "Zhi" in "Zhi Qian" can also be interpreted as the common word meaning "'s" or "of". The comma predictor (correctly) predicts commas before and after this name, and the name tagger then recognizes it.

7. DISCUSSION

In summary, this work found that automatically predicted commas, despite a relatively low F-measure, can lead to a significant improvement in both POS and name tagging, relative to the case of using only automatically predicted sentence boundaries in ASR transcripts. It is possible to distinguish between comma and caesura in automatic prediction, in that there are few confusions between the two types, but it did not lead to significant gains in POS or name tagging.

While the precision for comma prediction is high, recall is relatively low because of false predictions of sentence boundaries at comma locations. Joint prediction of commas and sentence boundaries would likely improve performance, though it is not clear that it would have a big impact on tagging. Perhaps a better area for potential gains would be joint modeling of punctuation and tagging.

8. REFERENCES

[1] J. Makhoul *et al.*, "The effects of speech recognition and punctuation on information extraction performance," in *Proc. Eurospeech*, 2005, pp. 57–60.

[2] A. Stolcke *et al.*, "Recent innovations in speech-to-text transcript at SRI-ICSI-UW," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1729–1744, 2006.

[3] M. Huang *et al.*, "Investigation on Mandarin Broadcast News Speech Recognition," in *ICSLP*, 2006.

[4] S. M. Thede and M. P. Harper, "A second-order hidden Markov model for part-of-speech tagging," in *Proc. ACL*, 1999, pp. 175–182.

[5] M. Collins and T. Koo, "Discriminative reranking for natural language parsing," *Computational Linguistics*, vol. 31, no. 1, pp. 25–70, 2005.

[6] D. Bikel *et al.*, "Nymble: A high-performance learning namefinder," in *Proc. Conference on Applied Natural Language Processing*, 1997, pp. 194–201.

[7] E. Shriberg *et al.*, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, no. 1-2, pp. 127–154, 2000.

[8] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech," in *Proc. ICSLP*, 2002, pp. 917–920.

[9] Y. Liu *et al.*, "Enriching speech recognition with sentence boundaries and disfluencies," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1526–1540, 2006.

[10] D. Beeferman *et al.*, "CYBERPUNC: A lightweight punctuation annotation system for speech," in *Proc. ICASSP*, 1998.

[11] Z. Zhang *et al.*, "Intra-sentence punctuation insertion in natural language generation," Tech. Rep. MSR-TR-2002-58, Microsoft Research, 2002.

[12] M. Zimmerman *et al.*, "The ICSI+ multilingual sentence segmentation system," in *Proc. Interspeech*, 2006.

[13] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[14] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *ICSLP*, 2002, vol. 2, pp. 901–904.

[15] A. Venkataraman *et al.*, "An efficient repair procedure for quick transcriptions," in *Proc. ICSLP*, 2004, pp. 2002–2005.