PUNCTUATING SPEECH FOR INFORMATION EXTRACTION

Benoit Favre[1]*, Ralph Grishman*[2]*, Dustin Hillard*[3]*, Heng Ji*[2]*, Dilek Hakkani-Tür*[1]*, Mari Ostendorf*[3]

[1]ICSI, 1947 Center St, Suite 600, Berkeley, CA 94704, USA, {favre,dilek}@icsi.berkeley.edu
[2]New York University, New York, NY 10003, USA, {grishman,hengji}@cs.nyu.edu
[3]SSLI Lab., EE Dept., University of Washington, Seattle, WA, {hillard,mo}@ee.washington.edu

ABSTRACT

This paper studies the effect of automatic sentence boundary detection and comma prediction on entity and relation extraction in speech. We show that punctuating the machine generated transcript according to maximum F-measure of period and comma annotation results in suboptimal information extraction. Precisely, period and comma decision thresholds can be chosen in order to improve the entity value score and the relation value score by 4% relative. Error analysis shows that preventing noun-phrase splitting by generating longer sentences and fewer commas can be harmful for IE performance. Indeed, it seems that missed punctuation allows syntactic parsers to merge noun-phrases and prevent the extraction of correct information.

Index Terms— Speech, Punctuation Prediction, Information Extraction

1. INTRODUCTION

Information Extraction (IE) aims at finding semantically defined entities in documents and characterizing relations between them. This task is a fundamental step in coping with today's *information overload* since it brings Natural Language Processing (NLP) to a higher level of understanding. IE outputs are used as features in various tasks like machine translation, summarization or information distillation [1].

Advances in speech processing make the application of IE possible on spoken documents, beyond the traditional textual documents. This new media involves many difficulties related to its variability in terms of quality, environment, speaker and language. Moreover, IE is generally applied on top of machine generated transcription and automatic structuring that suffer from errors compared to the true content.

Having been developed on textual content, IE presupposes properties like punctuation to be available. Unlike text where punctuation is usually explicit, punctuating speech can be hard due to disfluencies, incomplete sentences, hesitations, etc. Sentence segmentation is required in order to limit the processing of syntactic parsers, and sentence-internal punctuation also seem important. For instance, [2] observed that removing commas affects IE performance. Moreover, commas improve Chinese name recognition and part of speech tagging [3].

In this paper, we hypothesize that IE on speech can be improved by generating punctuation with IE performance in sight instead of the accuracy of the punctuation itself. Unlike [2], we generate both periods and commas automatically and examine their effect on IE.

After presenting the experimental setup in Section 2, we show in Section 3 that:

1. Maximizing the F-measure of punctuation detection does not result in maximal performance for IE.

2. Different thresholdings of punctuation output are required for entity and relation detection.

Section 4 discusses the behavior of the system with a deeper analysis of the results. Section 5 summarizes the contributions and discusses future work.

2. EXPERIMENTAL SETUP

2.1. Data

All the presented experiments are conducted on the portion of TDT4 English broadcast news that overlaps with the ACE'04 (Automatic Content Extraction) information extraction reference data. The speech has been transcribed by SRI's Broadcast News speech recognizer [4] with an estimated word error rate of 18% (the reference is formed by subtitles that do not exactly match the real discourse). In total, we use 131 stories from 101 shows that represent approximately 38k words and 4 hours of speech. Mean sentence length in the reference data is close to 15 words.

This set of stories is split into a test set for evaluation and a development set (dev) for parameter tuning. Each of the two sets represent half of the data. Punctuation prediction and information extraction systems are trained on separate data from similar corpora as detailed in the next section. Automatic story segmentation is not considered in this work and reference story boundaries are used.

2.2. Punctuation Prediction

Since the notion of a sentence is very different in speech compared to written text, we focus on only two types of punctuation: periods and commas. Experiments showed that joint prediction of commas and periods was similar to independent prediction, so we chose to model them separately in order to investigate different thresholds over the two types of events.

Speech is segmented into sentences by looking for likely sentence boundaries between consecutive words. The task is formalized as a binary classification problem using local features computed around potential boundaries. These features consist of lexical and part-of-speech ngrams, as well as the prosodic features described in [5]. Various groups of prosodic features are extracted: pause duration, speaker changes, pitch, energy and phone duration. Each is modeled on both sides of the boundary or compares the two sides. The features are normalized according to speaker and corpus-based statistics. Sentence-boundary events are predicted with a CRF sequence model similarly to [6]. Continuous features are quantized according to a boosted ensemble of threshold based decision stumps. This quantification scheme has proved to perform better than simple binning of continuous features since it also integrates the relationship of different features to each other regarding the classification objective. Sentence segmentation is traditionally evaluated using F-measure and NIST error rate [7]. F-measure, the harmonic mean of recall and precision on sentence boundary events, is used in this work to determine the sentence boundary decision threshold. The system is trained on 500k words from TDT4 (disjoint from the data used for IE evaluation). Decision thresholds are determined on the development set and evaluated on the test set.

Our comma modeling approach combines a hidden event language model (HE-LM) with word level posteriors from a boosted tree classifier (Boostexter) [3]. The word level classifier uses the same acoustic features that are used in our sentence boundary approach, as well as word ngram features. The Boostexter model is trained on a subset of TDT4 that is separate from the ACE data, with about 60k words. Reference commas are obtained by aligning commas from reference transcriptions to the words of the flexible alignment. Portions of the corpus with high alignment error (due to poor transcription) are removed from the training data. The number of iterations in training, as well as the optimal threshold, is tuned on a portion of held out development data. The HE-LM is a 5-gram with Kneser-Ney smoothing trained on a large collection of English text, including Gigaword, TDT2, TDT4 text, Hub4, Bizweek, and BBC text. Comma F-measure will be reported assuming true sentence boundaries.

2.3. Information Extraction

The IE components used for these experiments were developed for the ACE evaluations. The available speech data corresponded to documents used for the 2004 ACE evaluation, so we have used the 2004 ACE specifications and scoring rules throughout. ACE includes several separate tasks. *Entity Detection and Tracking* involves the identification of all entities in seven semantic classes (people, organizations, geo-political entities [locations with governments], other locations, facilities, vehicles, and weapons) which are mentioned in a document. In practice this involves finding *mentions* of entities (names, noun phrases, or pronouns), and then grouping mentions which refer to the same entity (coreference). *Relation Detection and Characterization* involves finding specified types of semantic relations between pairs of entities. For 2004 evaluations, there were 7 types of relations and 23 subtypes, including a *located-in* relation, *employment* relations, a *citizen-of* relation, and a *subsidiary-of* relation.

Entity detection and tracking involves several separate processing components. Names are identified and classified using an HMM-based name tagger trained on several years of ACE data. Noun groups are identified using a maximum-entropy-based chunker trained on part of the Penn TreeBank, and then semantically classified using statistics from the ACE training corpora. Coreference is rule based, with separate rules for name, nominal, and pronominal anaphors. For relation detection, we classify each relation in the training corpus based on the type and heads of the arguments, and selected words appearing between the arguments, and define a distance metric based on these features. Relations in new sentences are then identified using a nearest-neighbor procedure. The name model was trained on 800K words; the nominal classifier on 600K words, and the relation model on about 90K words of ACE training data. More details about the IE system can be found in [8].

3. RESULTS

Three kinds of experimental setups are evaluated in this section: baselines, a system where punctuation is optimized toward its own performance, and a system where punctuation prediction is tuned in order to optimize IE. All IE results are given in terms of the entity value and relation value scores, as produced by the official ACE 2004 scorer. These value scores include weighted penalties for missing items, spurious items, and for feature errors in corresponding items; details are given in the ACE 2004 Evaluation Plan[1]. Scoring is based on offsets in the reference text, so after IE is performed, offsets in the STT output are mapped (based on a token alignment) into offsets in the reference text and the result is then scored.

The baselines compare the effect of reference punctuation (upper bound) and pause-based punctuation (lower bound) on IE annotation. The reference punctuation is restricted to periods and commas (and periods only) while the pause-based punctuation corresponds to segmenting the word stream at pauses greater than 70ms (giving a period detection

[1] http://www.itl.nist.gov/iaui/894.01/tests/ace/ace04/doc/ace04-evalplan-v7.pdf

F-measure of 50.0). A no-boundary baseline would generate document-length sentences that can provoke system failure. In addition, the effect of STT is compared to the reference words. Note that the reference words are flexible alignments from the closed captions stripped from case information. They do not reflect the performances of the IE system on text data since the ACE reference has been designed on the closed-captions instead of the spoken words. Results are presented in Table 1, showing that both erroneous words and poor punctuation affect IE and that the effect is cumulative. Interestingly, entity score is lower on full punctuated words compared to period-only punctuated words. Though this is likely to be an effect of flexible alignment and STT, further investigation is needed.

Words	**Punctuation**	**Entity**	**Relation**
Ref	Reference	55.7	22.1
Ref	Ref. w/o commas	56.3	20.0
Ref	Pause-based	54.9	18.8
STT	Reference	46.9	20.0
STT	Ref. w/o commas	47.3	16.5
STT	Pause-based	47.0	15.6

Table 1. Baseline IE performance on entities and relations (on the test set). Various conditions are presented: reference words (Ref), machine generated words (STT), reference punctuation (with and without commas) and pause-based segmentation (pauses $>$ 70ms).

	Opt.	**thr$_p$**	**thr$_c$**	**F$_p$**	**F$_c$**	**Ent.**	**Rel.**
Dev.	Punc.	0.27	0.68	**68.5**	**41.2**	43.6	10.9
	Ent.	0.09	0.50	59.8	39.6	**46.0**	12.8
	Rel.	0.21	0.28	67.7	37.8	43.8	**14.1**
Test	Punc.	0.27	0.68	**65.1**	40.2	46.1	17.6
	Ent.	0.09	0.50	58.0	**40.7**	**48.2**	16.9
	Rel.	0.21	0.28	64.1	39.8	46.1	**18.4**

Table 2. Comparing IE performance when punctuation is self-optimized (Punc.) or optimized in order to improve entities (Ent.) and relations (Rel.). Period and comma decision thresholds (thr$_p$, thr$_c$) are chosen in order to maximize performance on the development set and used blindly on the test set. Punctuation F-measure is reported in F$_p$ and F$_c$.

In order to verify the hypothesis that traditional maximization of punctuation classification performance is suboptimal for the task of IE, the system is run for two conditions on STT words. First, IE is performed on punctuation decisions resulting from maximizing F-measure for period and comma annotation on the development set. Then, the decision thresholds are chosen according to IE performance on the same set. The results are reported in Table 2 for the development and test sets and show that optimizing punctuation for IE is valuable because the entity score on the test set can be improved by 4% relative (significance level: $p < 0.06$) and the relation score can be improved by 4% relative ($p < 0.01$). But, the best setup for one score is not optimal for the other one. A drawback of this finding is that a single punctuation output will not suit both tasks; perhaps a sentence boundary value for each token will need to be passed to IE.

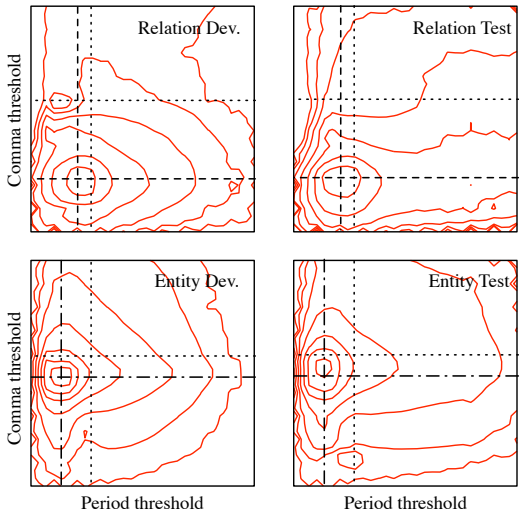

Fig. 1. IE performance on entities and relations when period and comma thresholds are varied from 0 to 1 (from left to right and bottom to top). Contours are displayed every 0.2 point drop from the highest score (artifacts are created by undersampling). The punctuation-optimal thresholds are indicated by dotted lines; the entity-optimal thresholds by dash-dot lines; the relation-optimal thresholds by dashed lines.

Figure 1 shows IE performance according to comma and period posterior probabilities. An interesting result is that development and test conditions lead to quite similar parameter spaces, which is favorable for the robustness of IE-oriented optimization. The plots also make it easy to find a good compromise between optimal entity and relation scores.

4. DISCUSSION

IE may be influenced by very short or very long sentences (one-word long to document-long). Intuitively, shorter sentences are more likely to break entities, especially if they involve long phrases. However, it is not obvious why fewer sentence boundaries would decrease the IE score. Therefore, we studied the output of the system and observed that one major effect of missed punctuation was noun-phrase (NP) merging. As illustrated by examples in Figure 2, if a sentence boundary is enclosed between NPs, it is likely that removing the period will confuse the NP chunker and merge the NPs. In this case, the former NP can act as adjunct to the head of the latter NP. Similarly, removing commas can lead to undetected appositions and erroneous parsing since in written text, the role of

the comma is often to disambiguate the syntactic parse. In our opinion, these errors are partly due to the assumed presence of punctuation when developing syntactic analysis rules. We observed that in ACE reference data, 28% of entity mentions (18% of heads) are adjacent to a punctuation mark.

A more subtle but significant effect relates to the fact that a sentence-initial token is more likely[2] to be a name than a sentence-internal token. In consequence, the HMM name tagger favors identifying tokens as names in sentence-initial position. In marginal cases, the tagger may correctly identify a name in sentence initial position (after a period) but miss it elsewhere. Having fewer periods, therefore, may lead to missed names and a lower entity value score.

(1)	*... aides [NP his children]. [NP senators] ...* *... aides [NP his children senators] ...*
(2)	*... the president of [NP mexico vincente fox]* *... the president of mexico, [NP vincente fox]*

Fig. 2. Examples where noun phrase assignment is ambiguous due to a missed sentence boundary (1) or comma (2). Even if semantically unlikely, the assignment is usually syntactically correct. Similarly, inserting a punctuation mark in the middle of a noun phrase will result in a split.

We also look at how punctuation over-generation affects entity mention splitting by computing the number of reference mentions split at different thresholds. Splitting a NP may result in two entity candidates from which at least one will affect performance (since heads are used for scoring). The curve in Figure 3 shows that choosing a lower threshold in order to reduce NP merging may actually result in more splits and not lead to IE improvement. The idea that shorter sentences are easier to annotate is only valid when the quality of the punctuation confidence scores is very high (we have observed that many true periods/commas have low confidence scores).

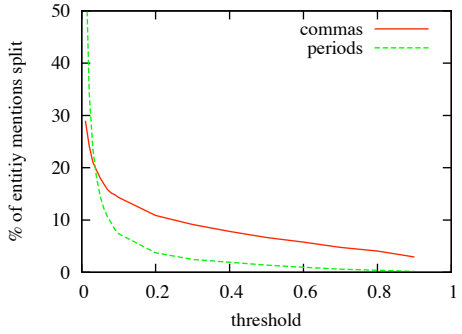

Fig. 3. Percentage of reference entity mention extents split by inserting commas or periods at their respective decision thresholds.

[2]For our training corpus, roughly twice as likely.

5. CONCLUSION

The work presented in this paper focuses on adequately punctuating speech in order to improve information extraction (IE). Recovering punctuation for the STT transcript is necessary in order to take advantage of the annotated textual data available in larger quantity than speech data, to train IE, like the majority of high level natural language processing tasks. We have shown that setting the punctuation decision thresholds to maximize punctuation performance is sub-optimal for IE. Moreover, improvements are obtained at different thresholds when annotating entities or relations. This suggests that punctuation should be generated differently according to the final aim. An analysis of the results showed that punctuation errors can result in merged noun phrases or split entities. The former phenomenon can be traced to syntactic parsing that usually requires accurate punctuation. As future work, we suggest improving the integration of speech related parameters in IE by, for example, optimizing STT for parsing performance or adapting the parser to ill-punctuated content. Eventually, we would like to introduce new features in punctuation prediction, inferred from IE output.

6. REFERENCES

[1] M. Levit, D. Hakkani-Tür, and G. Tur, "Integrating Several Annotation Layers for Statistical Information Distillation," in *Proc. of ASRU*, 2007.

[2] J. Makhoul, A. Baron, I. Bulyko, et al., "The Effects of Speech Recognition and Punctuation on Information Extraction performance," in *Proc. of Interspeech*, 2005.

[3] D. Hillard, Z. Huang, H. Ji, et al., "Impact of Automatic Comma Prediction on POS/Name Tagging of Speech," in *Proc. of SLT*, 2006.

[4] A. Venkataraman, R. Gadde, A. Stolcke, D. Vergyri, et al., "SRIs 2004 Broadcast News speech to text system," in *Proc. of Fall RT'04 Workshop*, 2004.

[5] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tur, "Prosody-Based Automatic Segmentation of Speech into Sentences and Topics," *Speech Communications*, 2000.

[6] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *Proc. of ACL'05*, 2005, pp. 451–458.

[7] Y. Liu and E. Shriberg, "Comparing Evaluation Metrics for Sentence Boundary Detection," in *Proc. of ICASSP*, 2007.

[8] R. Grishman, D. Westbrook, and A. Meyers, "NYU's English ACE 2005 System Description," in *Proc. of ACE 2005 Workshop*, 2005.