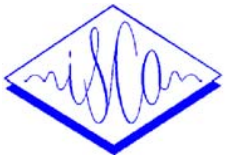
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

The ICSI-SRI-UW Metadata Extraction System

Yang Liu[1,4] *Elizabeth Shriberg*[1,2] *Andreas Stolcke*[1,2] *Dustin Hillard*[3]
Mari Ostendorf[3] *Barbara Peskin*[1] *Mary Harper*[4]

[1]International Computer Science Institute, USA [2]SRI International, USA
[3]University of Washington, USA [4]Purdue University, USA
{yangl,ees,stolcke,barbara}@icsi.berkeley.edu, {hillard,mo}@crow.ee.washington.edu

Abstract

Both human and automatic processing of speech require recognizing more than just the words. We describe a state-of-the-art system for automatic detection of "metadata" (information beyond the words) in both broadcast news and spontaneous telephone conversations, developed as part of the DARPA EARS Rich Transcription program. System tasks include sentence boundary detection, filler word detection, and detection/correction of disfluencies. To achieve best performance, we combine information from different types of language models (based on words, part-of-speech classes, and automatically induced classes) with information from a prosodic classifier. The prosodic classifier employs bagging and ensemble approaches to better estimate posterior probabilities. We use confusion networks to improve robustness to speech recognition errors. Most recently, we have investigated a maximum entropy approach for the sentence boundary detection task, yielding a gain over our standard HMM approach. We report results for these techniques on the official NIST Rich Transcription metadata tasks.

1. Introduction

Although speech recognition technology has improved significantly in recent decades, current speech systems still output simply a "stream of words". Unlike written text, this unannotated word stream leaves out useful information about punctuation and disfluencies. Such structural information is important for human readability of speech transcripts [1]. It is also crucial to applying downstream natural language processing techniques, which are typically based on the assumption of fluent, punctuated, and formatted input. Recovering structural information in speech has thus become the goal of a growing number of studies in computational speech processing [2, 3, 4, 5, 6, 7].

To this end, the metadata extraction (MDE) research effort within the DARPA EARS program (see http://www.darpa.mil/ipto/programs/ears/) aims to enrich speech recognition output by adding automatically tagged information on the location of sentence boundaries, speech disfluencies, and other phenomena. In this paper we focus on metadata encoding structure at the word level, and will not touch on speaker change detection and labeling, which are also part of the broader MDE effort.

In this paper, we describe the ICSI-SRI-UW metadata extraction system, which yielded the best performance on most MDE tasks in the most recent NIST 2003 Fall MDE evaluation. We introduce the MDE tasks and scoring approach in this section. Section 2 describes our basic system, including the knowledge sources and modeling techniques employed. Section 3 shows the results using confusion networks for the SU[1] detection task. Section 4 describes our recent investigation of the maximum entropy modeling for detecting SUs. Conclusions appear in Section 5.

[1]SU stands for sentence-like units; see LDC's annotation guidelines [8] for the definition of an SU.

1.1. MDE Tasks

The Rich Transcription structural MDE framework includes four tasks.

- "Sentence unit" (SU) detection aims to find the end point of an SU. SUs correspond to either complete or incomplete sentences.
- "Edit word" detection aims to find all words within the reparandum region of a speech repair, or the word region that when deleted yields a "fluent" version of the utterance.
- "Filler word" detection aims to identify words used as filled pauses (FP) or discourse markers (DM).
- "Interruption point" (IP) detection aims to find the inter-word location at which point fluent speech becomes disfluent.

The following example shows a transcript with metadata marked: '/' for SU boundaries, '< >' for fillers, '[ ]' for edit words, and '*' for IPs.

```
and  < uh > < you know > wash your clothes
wherever you are / and [ you ] * you really
get used to the outdoors /
```

Each task is evaluated separately. Systems are evaluated on both reference (human) transcriptions and the output of an automatic speech recognition system. Scoring tools first align the reference and hypothesis words, then map metadata events, and then calculate the errors. For the edit and filler word detection, the errors are the average number of misclassified reference tokens per reference edit or filler word token. For SU and IP detection, the errors are the number of misclassified points (missed and falsely detected points) per reference SU or IP. When recognition output words do not align perfectly with those in reference transcripts, an alignment that minimizes the word error rate is used and then the hypothesized metadata events are mapped to the reference metadata events. Further description is provided in http://www.nist.gov/speech/tests/rt/rt2003/fall/.

1.2. MDE Corpora

Evaluation is performed on two corpora that differ in speaking style: conversational telephone speech (CTS) and broadcast news (BN). Training and test data are those used in the DARPA Rich Transcription Fall 2003 evaluation.[2] The CTS data set contains roughly 40

[2]We used both the development set and the evaluation set as the test set in this paper, in order to increase the test set size to make the results more meaningful.

10.21437/Interspeech.2004-222

hours of speech for training and 6 hours (72 conversations) for testing. The BN data contains about 20 hours for training and 3 hours (6 shows) for testing. Training and test data are annotated with metadata events by LDC, using guidelines detailed in [8].

2. Baseline System and Performance

2.1. Previous Framework

Boundary detection problems may be viewed as classification tasks. In the training data, SU boundaries and IPs are marked by annotators using both the information in the transcription and the recorded speech. For testing, given a word sequence (human transcription or speech recognition output) $W_1\ W_2\ ...\ W_n$ and the speech signal, we use various knowledge sources (e.g., prosody and lexical information) to determine whether a given inter-word boundary should be a marked event (SU boundary or IP) or a nonevent.

Our boundary classifier has three components: the prosody model, the hidden event language model (LM), and various strategies for combining these models [9]. The prosody model is a probabilistic classifier that estimates the conditional probability of a boundary class at each word boundary, given features associated with that boundary. The features reflect prosodic patterns, including duration, fundamental frequency (F0), energy and pause. We chose a decision tree classifier to implement the prosody model.

A hidden event LM [10] models the joint distribution of boundary types and words. For a sequence $W_1\ E_1\ W_2\ E_2\ ...\ W_n$, where the metadata events E_i are included as pseudo-word tokens, the LM models the joint probability of the word and event sequence. Standard N-gram modeling techniques can be applied to implement the hidden event LM.

The most successful integration approach from our past work is based on a hidden Markov model (HMM) defined by the transition probabilities given by the hidden event LM and observation probabilities estimated by the prosodic model. Posterior probabilities $P(E_i|F_i)$ estimated by the prosodic decision tree are converted to likelihoods $P(F_i|E_i)$ for this purpose. Thus the integrated HMM models the joint distribution $P(W, F, E)$ of word sequence W, prosodic features F, and the hidden event sequences E. Standard algorithms are then applied to extract the most probable event type at each inter-word location $\hat{E}_i = \text{argmax}_{E_i} P(E_i|W, F)$, given the word sequence W and the prosodic features F.

2.2. System Description

Based on the general approach described above, we enhanced the language modeling of the system with a part-of-speech (POS) based hidden event LM, a hidden event LM based on automatically induced word classes, and a repetition detection LM [11]. POS tags are obtained from TnT taggers [12], trained using the Switchboard Treebank data and the broadcast news corpus. Automatically induced classes are obtained using the algorithm described in [13]. Additionally, we have a large Broadcast News recognizer LM that is trained from a large text corpus [15]. These various hidden-event LMs are combined via linear interpolation.

For the prosody model, in order to address the imbalanced data problem (since there are many fewer metadata events than nonevents at inter-word boundaries), we use a downsampled training set. Additionally, we employ ensemble bagging to reduce the variance of the prosodic classifier. In this method, several downsampled training sets are generated, and each is resampled multiple times and corresponding classifiers are combined via bagging [14]. This substantially improves the performance of the prosody model.

We build separate two-way classifiers for each task: SU vs. non-SU, edit IP vs. non-IP, FP vs. non-FP, DM vs. non-DM. During testing, the prosody model and multiple LMs are combined to obtain the best hypothesis for each inter-word boundary. For edit word detection, we use the IP hypotheses and work backwards, looking for words that match the word following the IP.

Since we use separate classifiers for each task, there are possibly conflicts between different classifiers' decisions at an inter-word boundary. We reconcile the SU and edit IP decision conflict by looking at the posterior probability of SU detection (which is more accurate than the IP classifier); when it is higher than a predefined threshold, the SU hypothesis is preserved; otherwise the IP hypothesis is used. Hypothesized IPs are also added at the beginning of filler words in a post-processing step.

2.3. System Performance

Table 1 shows system performance for all structural metadata tasks on both BN and CTS, and using both reference transcription (REF) and speech recognition output (STT). STT output is obtained from the SRI recognizer [15], with a word error rate of 12.1% on BN and 22.9% on CTS.

	BN		CTS	
	REF	STT	REF	STT
SU	48.72	55.37	31.51	42.97
Edit	51.37	100.39	59.22	87.99
Filler	9.22	52.45	18.07	47.97
IP	17.51	74.47	27.13	65.75

Table 1: System performance (error rate in %) for all the structural MDE tasks on CTS and BN test sets.

As shown, performance degrades dramatically in the face of recognition errors for all the tasks. Note, however, that the degradation on the SU detection task is less than on other tasks, which could be due to several reasons. For one thing, the prosody model, which is more robust to recognition errors, was found to be especially effective for the SU task. Also, the language model for SUs is not as dependent on just a few key words or patterns as in the case of filler word detection or disfluencies (which are cued by repeated words). For edit disfluencies and IPs the reference condition provides word fragments, which constitute very reliable cues, but are completely absent in automatic recognition output. Finally, SU events have a higher frequency than the other metadata events, thereby making model estimation relatively more robust for this task.

2.4. Contributions of Knowledge Sources

Table 2 presents the contributions from different components for SU detection for the two corpora. We focus here and in all further sections on only the SU task, due to space constraints. The SU task is a good choice, because unlike the disfluency tasks, SU events are frequent in both CTS and BN data.

As shown, performance improves as knowledge sources are added. A better prosody model using ensemble bagging ('prosody-ens-bag' in the table) generates better posterior probabilities given the prosodic features, and thus combines better with LMs, than using a single downsampled training set ('prosody-ds' in the table). Class-based LMs (POS and automatically induced classes) provide some additional gain when combined with the word-based LM, presumably by addressing the sparse data problem and by capturing some amount of syntactic or semantic information. Also apparent from the table is the finding that word recognition errors degrade the LMs relatively more than they degrade the prosody model. Using recognized rather than true words is more of a problem for CTS

	BN		CTS	
	REF	STT	REF	STT
word LM	68.16	72.54	40.56	51.85
word LM + prosody-ds	53.61	59.69	35.05	45.30
word LM + prosody-ens-bag	50.03	56.17	32.71	43.71
prosody-ens-bag	72.94	72.09	61.23	64.35
word + POS+ class LM + prosody-ens-bag	48.72	55.37	31.51	42.97

Table 2: Contributions of components for SU detection for both CTS and BN tasks, REF and STT conditions. Results are shown in error rate (%).

than for BN; this is most likely attributable to the word error rates on CTS, which are about twice as high as for BN recognition.

3. Confusion Networks

The significant degradation in performance on MDE tasks when using the best recognizer hypothesis (versus the true words) motivates an approach that can integrate information from multiple word hypotheses. Multiple word hypotheses are valuable because while the top recognizer output is optimized to reduce word error rate, alternative hypotheses may together reinforce alternative (more accurate) predictions of metadata events. We focus on CTS because of its relatively higher WER.

In recent work, we have examined the use of confusion networks [16] to leverage multiple recognizer hypotheses in predicting SUs. For each hypothesized word sequence, an HMM is used to estimate the posterior probability of an SU at each word boundary. The hypotheses are combined using confusion networks to determine the overall most likely event at each boundary [17].

Table 3 shows SU error rates for a system that includes bagged trees and an interpolated class LM. The system combines hypotheses from a pruned n-best list that utilizes the top 90% of the recognizer hypotheses (by posterior mass) in order to limit the processing time required. System performance on the single best hypothesis from this pruned list is given for comparison.

Single Best	Single Best (pruned)	Confusion Nets
43.62	44.29	43.11

Table 3: SU error rates (%) for 1-best recognition versus the confusion network approach for CTS with WER 22.9%.

Combining predictions from multiple hypotheses reduces error rates relative to 1-best predictions. Gains compared against the single best system are smaller than against the pruned single best, because only a portion of the hypotheses are used in the latter. Producing hypotheses for the entire n-best list may lead to further gains. In ongoing work we are moving from n-best confusion networks to a lattice framework, allowing us to consider a much larger hypothesis space.

4. Maximum Entropy Modeling

A weakness of the HMM-based model combination approach described earlier is that it assumes independence of lexical (N-gram) and prosodic features given the metadata events. We have therefore begun to explore maximum entropy (maxent) models as an alternative approach to feature integration. Here we report on maximum entropy modeling for the SU detection task.

4.1. Maxent Model Description

The maxent model for SU boundary detection assigns a posterior probability for SU boundary at each inter-word boundary, given the features associated with each boundary. The maxent estimator finds a model that satisfies all feature expectation values derived from the training data, while being maximally smooth (i.e., having maximum entropy). This model has the following exponential form:

$$p(y|x) = \frac{1}{Z(x)} exp(\sum_i \lambda_i f_i(x,y)) \quad (1)$$

where $Z(x) = \sum_y exp(\sum_i \lambda_i f_i(x,y))$, $f_i(x,y)$ is the indicator function for feature i, y is the metadata event type, and x represents the context associated with the sample.

Maxent modeling has been employed in many natural language processing tasks [18]. For a language processing task, the features are generally easy to define, and mostly characterize the context of an event. The power of the maxent approach stems from the fact that multiple features can apply to the same event, without having to model explicitly the joint occurrence of such features.

For SU detection we utilize both textual features and features derived from the prosody model.

Word: We use various combinations of word contexts to represent word features. The features include different lengths of N-gram and different positional information for a location i, e.g., $<w_i>$, $<w_{i+1}>$, $<w_i, w_{i+1}>$, $<w_{i-1}, w_i>$, $<w_{i-2}, w_{i-1}, w_i>$, and $<w_i, w_{i+1}, w_{i+2}>$.

POS: POS tags are the same as used for the HMM approach. Features capturing POS information are similar to those used for words.

Chunk: Chunks are obtained from a TBL chunker trained on the Wall Street Journal corpus [19]. Each word has an associated chunk tag, such as the beginning of an NP, inside a VP, etc. We use the same combination of contexts for chunk tags as used for word and POS tags. This type of feature is used only on the BN task because of the poor chunking performance on CTS.

Class: We also use similar features coming from automatically induced classes.

Turn: Since speaker changes are very indicative of SU boundaries, we use this binary feature indicating speaker change.

Prosody: To keep the prosodic classifier as a separate model component, and since the maxent classifier is most conveniently used with binary features, we encode the posterior probabilities from the prosodic decision tree into several binary features through thresholding. Equation (1) shows that the presence of each feature in a maxent model has a monotonic effect on the final probability (raising or lowering it by a constant factor). It is therefore best to define binary features encoding the decision tree posterior probabilities p in a cumulative fashion: $p > 0.1, p > 0.3, p > 0.5, p > 0.7, p > 0.9$, with heuristically chosen thresholds. This representation is also more robust to the mismatch between the posterior probability in training and test sets, since small changes in the posterior value affect at most one feature.

LM: It is convenient to include posterior event probabilities from additional LMs (obtained using the HMM framework), rather than encoding the LM information as a large number of features in training. This is especially attractive for LMs trained from text-only sources, such as the large Broadcast

News recognizer LM. The LM posterior probabilities are encoded as binary features similar to the decision tree posteriors.

To date, we have not fully investigated compound features that combine different knowledge sources and are able to model the interaction between them explicitly. We included only a limited set of such features, such as the combination of the decision tree's hypothesis and POS contexts.

4.2. SU Results Using Maxent

Table 4 shows SU detection results using maxent and HMM approaches individually, as well as their combination. The combination is carried out by a simple interpolation of posteriors from the two models. We observe that on the REF condition, for both BN and CTS, maxent achieves slightly better performance than HMM, and worse results on the STT condition. Maxent better models the overlapping textual information. However, the prosodic information is incorporated only through the rough thresholding; therefore, prosody information is not completely preserved and is under-used in the maxent approach. Because of the different errors made by the two approaches, the combination of maxent and HMM yields the best performance for all the test conditions.

		HMM	Maxent	Combination
BN	REF	48.72	48.61	46.79
	STT	55.37	56.51	54.35
CTS	REF	31.51	30.66	29.30
	STT	42.97	43.02	41.88

Table 4: SU detection results (error rate in %) using maxent and HMM individually, and their combination.

5. Conclusions

We have described various knowledge sources and modeling approaches for automatic detection of metadata events in both conversational and broadcast news speech. Our HMM-based system achieved state-of-the-art results in a recent government-sponsored evaluation. Prosodic model performance was greatly improved by using sampling and ensemble techniques to make better use of inherently skewed data. Different class-based language models yielded an additional gain beyond a word-based language model alone.

We have also explored new modeling techniques to try to address two types of problems. To address the problem of errorful speech recognition output, we have explored the use of multiple recognition outputs for finding locations of likely metadata events. An approach based on confusion networks has shown improvements on CTS recognition output. To better account for feature dependence across models, we have begun to investigate maximum entropy modeling. This approach outperforms our previous HMM-based integration on the reference condition of both BN and CTS. Furthermore, combining the system output from both the maxent and the HMM approach achieves the best performance across all the test conditions.

6. Acknowledgments

This research has been supported by DARPA under contract MDA972-02-C-0038, NSF-STIMULATE under IRI-9619921, ARDA under MDA904-03-C-1788, and NASA under NCC 2-1256. Distribution is unlimited. Any opinions expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA, NSF, ARDA, or NASA. Part of this work was carried out while the last author was on leave from Purdue University and at NSF.

7. References

[1] Jones, D., et al., "Measuring the Readability of Automatic Speech-to-Text Transcripts", Proc. Eurospeech, pp. 1585-1588, 2003.

[2] Heeman, P. and Allen, J., "Speech Repairs, Intonational Phrases and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialogue", Computational Linguistics, 25(4): 527-571, 1999.

[3] Kim, J. and Woodland, P. C., "The Use of Prosody In a Combined System for Punctuation Generation and Speech Recognition", Proc. Eurospeech, pp. 2757-2760, 2001.

[4] Gotoh, Y. and Renals, S., "Sentence Boundary Detection in Broadcast Speech Transcripts", Proc. ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000, pp. 228-235, 2000.

[5] Kompe, R., "Prosody in Speech Understanding System", Sringer-Verlag, 1996.

[6] Snover, M., Dorr, B., and Schwartz, R., "A Lexically-Driven Algorithm for Disfluency Detection", Proc. HLT/NAACL, pp. 157-160, 2004.

[7] Kim, J., Schwarm, S., and Ostendorf, M., "Detecting Structural Metadata with Decision Trees and Transformation-Based Learning", Proc. HLT/NAACL, pp. 137-144, 2004.

[8] Strassel, S., "Simple Metadata Annotation Specification V5.0", Linguistic Data Consortium, 2003.

[9] Shriberg, E., et al., "Prosody-based Automatic Segmentation of Speech into Sentences and Topics", Speech Communication, 32, pp. 127-154, 2000.

[10] Stolcke, A. and Shriberg, E., "Automatic Linguistic Segmentation of Conversational Speech", Proc. ICSLP, pp. 1005-1008, 1996.

[11] Liu, Y., Shriberg, E., and Stolcke, A., "Automatic Disfluency Identification in Conversational Speech Using Multiple Knowledge Sources", Proc. Eurospeech, pp. 957-960, 2003.

[12] Brants, T., "TnT-a Statistical Part-of-Speech Tagger", Proc. ANLP, pp. 224-231, 2000.

[13] Brown, P., et al., "Class-Based n-gram Models of Natural Language", Computational Linguistics, pp. 467-479, 1992.

[14] Liu, Y., et al., "MDE Research at ICSI+SRI+UW, NIST RT-03F Workshop", http://www.nist.gov/speech/tests/rt/rt2003/fall/presentations, 2003.

[15] Stolcke, A., et al., "Speech-To-Text Research at SRI-ICSI-UW, NIST RT-03S Workshop", http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/index.htm.

[16] Mangu, L., Brill, E., and Stolcke, A., "Finding consensus in speech recognition: word error minimization and other applications of confusion networks", Computational Linguistics, pp. 373-400, 2000.

[17] Hillard, D., et al., "Improving Automatic Sentence Boundary Detection with Confusion Networks" Proc. HLT/NAACL, pp. 69-72, 2004.

[18] Bergers, A., Della Pietra, S., and Della Pietra, V., "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics, pp. 39-72, 1996.

[19] Ngai, G. and Florian, R., "Transformation-Based Learning in the Fast Lane", Proc. NAACL, pp. 40-47, 2001.