

THE ICSI/SRI/UW RT04 STRUCTURAL METADATA EXTRACTION SYSTEM

Yang Liu^{1,3} *Elizabeth Shriberg*^{1,2} *Andreas Stolcke*^{1,2}
*Barbara Peskin*¹ *Mary Harper*³

¹International Computer Science Institute, USA ²SRI International, USA

³Purdue University, USA

{yangl,ees,stolcke,barbara}@icsi.berkeley.edu, harper@ecn.purdue.edu

ABSTRACT

Both human and automatic processing of speech require recognizing more than just the words. We describe the ICSI-SRI-UW metadata detection system 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 textual knowledge sources (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. In addition to our previous HMM approach, we investigate using a maximum entropy (Maxent) and a conditional random field (CRF) approach for various tasks. Results using these techniques are presented for the 2004 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, 8]. To this end, the metadata extraction (MDE) research effort within the DARPA EARS program 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 describe the ICSI-SRI-UW metadata extraction system, developed for the NIST RT-04 MDE evaluation. Our focus is on structural MDE in this paper, and will not touch on speaker diarization research. We introduce the MDE tasks and corpora in this section. Section 2 introduces the general approaches we used for the MDE tasks. Sections 3 through 6 describe the methods and show results for each of the MDE tasks. Conclusions appear in Section 7.

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. The detection of subtype (statement, backchannel, question, and incomplete) for each SU boundary is required in RT-04.
- “Edit word” detection aims to find all words within the reparandum region of a speech repair.
- “Filler word” detection aims to identify words used as filled pauses (FP), discourse markers (DM), and explicit editing terms.
- “Interruption point” (IP) detection aims to find the inter-word location at which point fluent speech becomes disfluent. This includes the starting point of a filler word string too.

Each task is evaluated separately. Systems are evaluated on both reference (human) transcriptions and the output of an automatic speech recognition system. System performance is measured by the number of misclassified metadata events per reference event. Detailed descriptions of the scoring metrics are provided in <http://www.nist.gov/speech/tests/rt/rt2004/fall/>.

1.2. MDE Corpora

Evaluation is performed on two corpora that differ in speaking style: conversational telephone speech (CTS) and broadcast news (BN). Test data for the RT-04 evaluation contains 3 hours (36 conversations) of CTS and 6 hours (12 shows) of BN.

LDC released the training data annotated according to the annotation guideline V6.2 [9]. There are about 40 hours of CTS data and 20 hours of BN data. The V6 guideline used for annotating these data is different from the V5 guideline used in RT-03 evaluation [10], with greater difference for CTS. We used only the RT-04 training data for most of the CTS MDE tasks, and merged the RT-04 and RT-03 BN data for the BN tasks. Since the annotation guideline did not change much for BN, and the data sparsity problem is more severe on BN, combining the two training sets increases the training data size and yields better performance, as observed from our preliminary experiments on the development sets. However, the differences between the two guidelines for CTS are large enough that using only the RT-04 training data, which matches better to the test set, is better. Our MDE models are trained using the reference transcriptions (plus speech data, and metadata annotation) and are applied to the reference and STT conditions in the same way.

For diagnostics, we evaluated on two different STT outputs for both BN and CTS. On CTS, we used SRI’s STT and IBM+SRI STT. On BN, we used SRI’s STT and the SuperEARS STT. The error rates of these STT results are shown in Table 1.

	STT	WER (%)
BN	SuperEARS	11.7
	SRI	15.0
CTS	IBM+SRI	14.9
	SRI	18.6

Table 1. WER of different STT outputs used in MDE evaluation.

2. GENERAL APPROACHES

We combined textual and prosodic features (reflecting duration, pause, pitch, and energy) for structural MDE tasks. For each interword boundary or each word, we use various features to determine whether there is a structural event at that boundary or whether that word belongs to an extent of a structural event. The general approaches (classifiers) used for various tasks are described in this section. The choice of the approach to use for a specific task and more detailed experimental setup will be described later for each task.

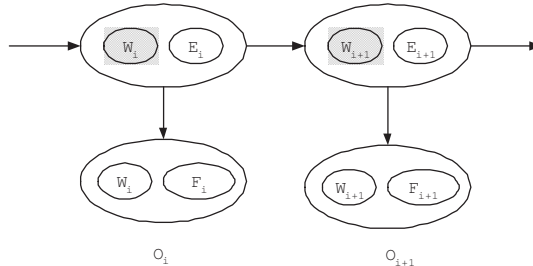


Fig. 1. The graphical model of an HMM for the MDE task. Only one word+event is depicted in each state, but in a model based on N-grams the previous $N - 1$ tokens would condition the transition to the next state.

2.1. Hidden Markov Model (HMM)

Our baseline model, and the one that forms the basis of much of the prior work on MDE [11, 4, 12, 3], is a hidden Markov model. Figure 1 shows the HMM approach for the structural event detection. The states of the model correspond to words w_i and following event labels e_i . Note that the words appear in both the states and the observations, such that the word stream constrains the possible hidden states to matching words; the ambiguity in the task stems entirely from the choice of events. A forward-backward algorithm is used to find the event with the highest posterior probability for each interword boundary:

$$\hat{E}_i = \operatorname{argmax}_{e_i} P(e_i|W, F) \quad (1)$$

where W and F are the words and prosodic features for the entire test sequence, respectively.

There are two sets of parameters to estimate. The state transition probabilities are estimated using a hidden event N-gram LM [13], which models the joint distribution of the word and event sequence $W, E = w_1, e_1, w_2, \dots, e_{n-1}, w_n$. The resulting LM can then compute the required HMM transition probabilities as:¹

$$P(w_i e_i | w_1 e_1 \dots w_{i-1} e_{i-1}) = P(w_i | w_1 e_1 \dots w_{i-1} e_{i-1}) \times P(e_i | w_1 e_1 \dots w_{i-1} e_{i-1} w_i)$$

The second set of HMM parameters are the observation likelihoods $P(f_i | e_i, w_i)$. Instead of training a likelihood model we make use of the prosodic classifiers. We have trained decision tree classifiers that estimate $P(e_i | f_i)$. If we further assume that prosodic features are independent of

¹To utilize word+event contexts of length greater than one we have to employ HMMs of order 2 or greater, or equivalently, make the entire word+event N-gram be the state.

words given the event type, observation likelihoods may be obtained by:

$$P(f_i|w_i, e_i) = \frac{P(e_i|f_i)}{P(e_i)}P(f_i) \quad (2)$$

Since $P(f_i)$ is constant we can ignore it when carrying out the maximization (1).

The HMM is a generative modeling approach since it describes a stochastic process with hidden variables (meta-data events) that produces the observable data. The HMM approach has two main drawbacks. First, the standard training methods for HMMs maximize the joint probability of observed and hidden events, as opposed to the posterior probability of the correct hidden variable assignment given the observations, which would be a criterion more closely related to classification performance. Second, the N-gram LM underlying the HMM transition model makes it difficult to use features that are highly correlated (such as word and POS labels) without greatly increasing the number of model parameters, which in turn would make robust estimation difficult.

2.2. Maximum Entropy (Maxent)

A maximum entropy (Maxent) posterior classification method has been evaluated in an attempt to overcome these shortcomings of the HMM approach [14]. Such a model takes the familiar exponential form:

$$P(e_i|W, F) = \frac{1}{Z_\lambda(W, F)} \exp\left(\sum_k \lambda_k g_k(e_i, W, F)\right) \quad (3)$$

where $Z_\lambda(W, F)$ is the normalization term. The functions $g_k(e_i, W, F)$ are indicator functions corresponding to features defined over events, words, and prosodic features. For example, one such feature function for the SU detection task might be:

$$g(e_i, W, F) = \begin{cases} 1 & : \text{if } w_i = \text{uhuh} \text{ and } e_i = \text{SU} \\ 0 & : \text{otherwise} \end{cases}$$

The Maxent model is estimated by finding the parameters λ_k with the constraint that the expected values of the various feature functions $E_{\mathcal{P}}[g_k(e', W, F)]$ match the empirical averages in the training data. These parameters ensure the maximum entropy of the distribution and also maximize the conditional likelihood $\prod_i P(e_i|W, F)$ over the training data. In our experiments we used the L-BFGS parameter estimation method, with Gaussian-prior smoothing [15] to avoid overfitting.

The conditional likelihood is closely related to the individual event posteriors used for classification, meaning that this type of model explicitly optimizes discrimination of correct from incorrect labels, which is an advantage of

the Maxent model over an HMM. Additionally, the Maxent framework provides a more principled way to combine a large number of overlapping features, as confirmed by the results of [14]; however, it uses only local information to make the decision for each boundary.

2.3. Conditional Random Field (CRF)

A simple combination of the Maxent and HMM was found to improve upon the performance of either model alone [14] for the SU detection task because of the complementary strengths and weaknesses of the two models. An HMM is a generative model, yet it is able to model the sequence via the forward-backward algorithm. Maxent is a discriminative model; however, it attempts to make decisions locally, without using sequential information. A conditional random field (CRF) model [16] combines the benefits of these two approaches.

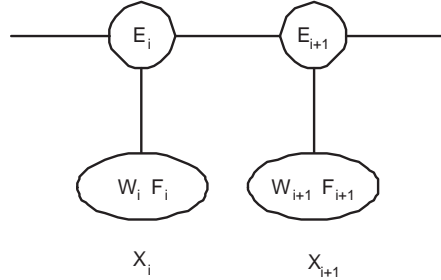


Fig. 2. A graphical representation of a CRF for the MDE problem. E represents the state tags (i.e., an event or not), while W and F are word and prosodic features respectively.

A CRF is a random field that is globally conditioned on an observation sequence X . CRFs have been successfully used for a variety of text processing tasks [16, 17, 18], but it has not been widely applied to speech related tasks with both acoustic and textual knowledge sources. Figure 2 is a graphical representation of this modeling approach. The states of the model correspond to event labels E_i . The observations X_i associated with the states are the words W_i , as well as other prosodic features F_i . The most likely sequence \hat{E} for the given input sequence (observations) X is:

$$\hat{E} = \underset{E}{\operatorname{argmax}} \frac{\exp(\lambda * G(E, X))}{Z_\lambda(X)} \quad (4)$$

where the function G is a potential function over the events and the observations, and Z_λ is the normalization term. The model is trained to maximize the conditional log-likelihood of a given training set. The most likely sequence is found

using the Viterbi algorithm.²

A CRF differs from an HMM with respect to its training objective function (joint versus conditional likelihood) and its handling of dependent word features. HMM training does not maximize the posterior probabilities of the correct labels; whereas, the CRF directly estimates posterior boundary label probabilities $P(E|W, F)$. The underlying N-gram sequence model of an HMM does not cope well with multiple representations (features) of the word sequence (e.g., words, POS); however, the CRF model supports simultaneous correlated features, and therefore gives greater freedom for incorporating a variety of knowledge sources. A CRF differs from the Maxent method with respect to its ability to model sequence information. The primary advantage of the CRF over the Maxent approach is that the model is optimized globally over the entire sequence; whereas, the Maxent model uses only local evidence (the surrounding word context and the local prosodic features), as shown in Equation (3).

We use the Mallet package [19] to implement the CRF model. To avoid overfitting, we employ a Gaussian prior with a zero mean on the parameters [15], similar to what is used for training Maxent models. The CRF takes longer to train than the HMM and Maxent models, especially when the number of features becomes large. The HMM requires less time for training than all the other models.

3. SU DETECTION TASK

For this task, we adopt a two-step approach. First we apply the SU boundary detection approach (HMM, Maxent, CRF, or some combination of these), then for each system hypothesized SU boundary, a classifier is used to determine its subtype. The reason we utilize a two-pass approach, rather than using the boundary detection approach with a 5-way classification (four SU subtypes plus non-SU), is to more easily incorporate knowledge about boundary locations (SU and SU initial words) into the subtype decisions.

3.1. SU Boundary Detection

We first describe model training and knowledge sources used in each approach, and then present experimental results in Section 3.3.

3.1.1. HMM

- *Prosody Model:* At each word boundary, we extracted prosodic features including duration, fundamental frequency (F0), energy, and pause [11]. A decision tree classifier is used as the prosody model

²The forward-backward algorithm would likely be better here, but it is not implemented in the current software used [19].

that generates the posterior probability of a meta-data event given the prosodic features at an inter-word boundary. To reduce the variance of a single decision tree, we employ bagging and ensemble approaches for prosody model training [20]. On CTS, we used bagging on a downsampled training set; on BN, we used ensemble bagging. These choices are based on the performance on the development sets and the computational efficiency considerations.

- *Textual Information:* The word identities themselves (from automatic recognition or human transcriptions) constitute a primary knowledge source for the SU task. We also make use of various automatic taggers that map the word sequence to other representations. The TnT tagger [21] is used to obtain part-of-speech (POS) tags. The tagged versions of the word stream are provided to allow generalizations based on syntactic structure and to smooth out possibly undertrained word-based probability estimates. For the same reasons we also generate word class labels that are automatically induced from bigram word distributions [22]. Similarly, we can interpolate LMs trained from different corpora. This is usually more effective than pooling the training data because it allows control over the contributions of the different sources. For example, we have a small corpus of BN training data labeled precisely to the LDC's SU specifications, but a much larger (130M word) corpus of standard broadcast news transcripts with punctuation, from which an approximate version of SUs could be inferred. The larger corpus should get a larger weight on account of its size, but a lower weight given the mismatch of the SU labels. By tuning the interpolation weight of the two LMs empirically (using held-out data) the right compromise was found.

In testing, for the state transition probability in the HMM approach, we combine the LMs from the hidden event word LM (trained from the LDC data and the extra corpus) and the automatically-induced class based LM. Since POS tags cannot be obtained on the fly, we adopted a loosely coupled approach to combine with the POS-based hidden event LM: the POS based LM is applied via the HMM approach (without using the prosody model) to generate posterior probabilities, which are then combined with the posteriors of the other models (various LMs and the prosody model).

3.1.2. Features Used in the Maxent and CRF

In the Maxent and CRF approaches, we utilize both textual information and features derived from the prosody model for SU boundary detection.

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., $\langle w_i \rangle$, $\langle w_{i+1} \rangle$, $\langle w_i, w_{i+1} \rangle$, $\langle w_{i-1}, w_i \rangle$, $\langle w_{i-2}, w_{i-1}, w_i \rangle$, and $\langle w_i, w_{i+1}, w_{i+2} \rangle$.

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 [23]. Each word has an associated chunk tag, such as the beginning of an NP, or inside a VP. 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 N-gram 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: 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 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. Incorporating the prosodic information in this way is convenient since the prosody model can be kept as a separate model component. The prosody model is the same as used in the HMM.

Additional LMs: 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.

3.2. SU Subtype Detection

After the SU boundary is detected, a second-pass is used to classify the boundary type. We use a Maxent classifier (rather than the HMM approach) for SU subtype detection because of the ease of incorporating various features such as sentence initial cue words. Features used include the SU initial words (after optional filler words), SU final words, the turn change information at the current and the previous SU boundaries, the length of the SU, and the binned posterior probabilities from a prosody model that does four-way SU subtype classification. For BN, we used the data for training the CTS SU subtype detection model, by removing the incomplete SUs from the training set and not using the prosodic features. Experiments on the development sets have shown that this yields better performance than using BN training data or CTS data with incomplete SUs preserved.

Table 2 shows the percentage of the four SU subtypes in CTS and BN data. For BN, statement is much more frequent than the other subtypes; whereas, for CTS the four types are more balanced, even though statement is still the majority. The highly skewed distribution of the subtypes on BN suggests that reasonably good performance can be achieved hypothesizing statement SU for every SU boundary hypothesis.

	statement	backchannel	question	incomplete
BN	94.23	0.86	4.37	0.53
CTS	62.05	26.80	5.11	6.05

Table 2. Percentage of SU subtypes for CTS and BN.

3.3. SU Detection Results

3.3.1. Eval Official Results

For the CTS SU detection task, we used a majority vote of the HMM, Maxent, and CRF approaches; for BN, a linear combination of the posterior probabilities from the HMM and Maxent approaches is used for generating the final decisions. After this step, a second-step subtype detection is conducted based on the boundary hypotheses. Table 3 shows our eval submission results for both BN and CTS.³ In the table, both the official error (including substitution error) and the boundary-only error are reported.

Clearly there is a large performance degradation for both CTS and BN on the STT condition compared to the reference condition. As expected, the poorer STT output leads to a greater error rate increase. In addition, the degradation is greater on CTS than on BN due to the higher WER on CTS.

³All the results reported in this paper are from md-eval-v17.pl.

Conditions		SU Error Rate (%)	Ins+Del
CTS	REF	36.80	26.21
	STT: IBM+SRI	49.24	39.18
	STT: SRI	54.12	44.26
BN	REF	49.71	47.15
	STT: SuperEARS	61.95	59.73
	STT: SRI	64.74	62.67

Table 3. Eval SU official results for BN and CTS, on REF and various STT conditions.

The SU error rate is generally higher on BN than CTS (especially when only accounting for the boundary detection results). This is partly because the performance is measured per reference SU event, and on BN the percentage of SU is smaller than on CTS. On the other hand, this also suggests that detecting SUs on BN is even harder than on CTS (relative to the chance performance in each domain).

3.3.2. Effect of Modeling Approaches

Table 4 shows the results for different modeling approaches on CTS data. Since these modeling approaches are used in the first pass of boundary detection, we report the boundary detection results (in parenthesis) in addition to the overall error rate. For the STT condition, results are reported using the better STT output (IBM+SRI). The contribution from each component in a single approach (such as different LMs, prosody model, etc) is not reported since the patterns are similar to our previous findings [24]. It can be seen from the table that the Maxent performs slightly better than the HMM, and CRF achieves the best performance among the three approaches when used alone. The results confirm that the CRF combines the advantage of the HMM and Maxent methods, and outperforms them. The CRF is even better than the combined results (via majority vote) on the reference condition. The toolkit we use for the implementation of the CRF does not have the functionality of generating a posterior probability for a sequence; therefore, we do not combine the system output via posterior probability interpolation, which we would expect to yield better performance.

	REF	STT (IBM+SRI)
HMM	39.46 (28.66)	50.70 (40.47)
Maxent	38.18 (27.59)	50.46 (40.27)
CRF	36.40 (26.27)	49.97 (40.27)
Combination	36.80 (26.21)	49.24 (39.18)

Table 4. CTS SU results using various modeling approaches, with the boundary detection error in parenthesis.

Table 5 shows the SU results for different modeling approaches on the BN SU task, for the reference and SuperEARS STT condition. The Maxent outperforms the HMM significantly on both the reference and the STT condition, and the combination of them yields the best performance. This gain is more than we have observed for the RT-03 data [14]. Additionally, the gain on the reference carries over to the STT condition, unlike in RT-03 data, where the Maxent suffers more from the word errors in the STT output. This may be partly because there are more data for Maxent model training (i.e., the combined RT-04 and RT-03 data).

	REF	STT (SuperEARS)
HMM	54.49 (51.76)	65.58 (63.34)
Maxent	52.09 (49.36)	63.26 (60.80)
Combination	49.71 (47.15)	61.95 (59.73)

Table 5. BN SU boundary detection results, with the boundary detection results in parenthesis.

3.3.3. Effect of Speaker Labels on BN

Speaker information is important for SU detection, since it is used in the prosodic features (as a separate feature and also used to derive features related to the length of a turn), used to generate turn change features, and it is also used to obtain per-speaker chunks for the hidden-event LM. This mostly affects BN where speaker information is unavailable; whereas, on CTS each channel corresponds to one speaker. In the results shown above for BN, we used the derived speaker labels from the speaker diarization results from the ICSI diarization system [25].⁴ For each pause-based segment, we find the speaker label from the speaker diarization results that has the majority speech in that segment. Another way to derive the speaker information is to use automatic clustering as is used in STT (e.g., for speaker adaptation and feature normalization). Table 6 compares the SU detection results using speaker information derived from these methods.

Approaches	SU Error
Use speaker diarization	54.49
Use automatic clustering in STT	64.10

Table 6. Comparison of different ways to derive speaker labels on BN SU task. Results are from the HMM approach.

We observe significant improvements when using the speaker information from the speaker diarization results,

⁴This result is from a preliminary system run on the eval data, which is slightly different from the final official diarization results.

suggesting that automatic speaker clustering may be better designed for speech recognition. The goal of automatic clustering used in STT is to cluster similar speakers together (based on acoustic similarity) for the purpose of recognizing words, and may not provide the correct speaker label.

3.3.4. Subtype Detection

We can see the SU subtype detection results in the previous tables. Note that in testing, features are extracted based on the system hypothesized SU boundaries; therefore, the starting point for a SU may be wrong (i.e., an SU detection insertion or deletion), which will affect the features related to SU initial words. Interestingly, as can be seen, generally the substitution errors are not much affected by the STT errors or the SU hypothesis errors. Recall also that the prosody model is built based on the features extracted around each word boundary; therefore, it does not account for the longer time prosodic features, which could be more useful for subtype detection.

4. FILLER WORD DETECTION TASK

Right now our system only detects filled pauses and discourse markers, not explicit editing terms. For filler word detection, we adopt the HMM boundary detection approach that first detects the filler word boundary, and then look backward to find the onset of the filler word string based on a list. Since filled pauses (FPs) are only single words, this is only applicable to discourse markers (DMs). We hypothesize that filled pauses and discourse markers are quite different phenomena; therefore, we use two separate models for FP and DM.

The hidden-event LMs are trained from the annotated transcription, to model the joint word and FP_END or DM_END sequence. The prosody model is trained from the downsampled training data, using the same prosodic features as used for the SU task. Our previous experiments have shown that the prosodic features used for filler detection are quite different from those for SU detection. Duration (e.g., word lengthening) is more important for filler detection than for SU.

	Conditions	Error Rate (%)
CTS	REF	27.10
	STT: IBM+SRI	42.53
	STT: SRI	44.64
BN	REF	18.11
	STT: SuperEARS	56.63
	STT: SRI	52.63

Table 7. Eval filler word detection submission results for BN and CTS, on REF and various STT conditions.

Table 7 shows the results for filler word detection. Notice from the table that the filler word detection performance is better when using the SRI STT output compared to the SuperEARS output on BN (mostly due to the filled pause detection). Since filled pauses are not provided in the SuperEARS output, we inferred their locations based on the SRI STT output, which was clearly suboptimal.

5. EDIT WORD DETECTION TASK

5.1. Methods

5.1.1. HMM

In our previous work, we used an HMM for edit IP detection, then applied heuristic rules for determining the start of edit disfluency [26]. The hidden event word-LM is trained from the joint word and editIP sequence. The prosody model is trained from the downsampled training set. Since the word-LM is generally undertrained and is able to detect only those repetitions that have occurred in the training set, we also use a repetition detection model, which finds the repeated word sequences with possible filler words allowed after the editIP [26].

5.1.2. Maxent

In the Maxent approach, we first use a Maxent classifier for SU/IP/NULL detection.⁵ Then similarly to the HMM approach, heuristic rules are used to determine the onset of reparandum. One advantage of this approach is that it jointly models the SU and IP events. For example, if ‘*that is great. that is great*’ has occurred in the training set, then the model will learn that these are two SUs, rather than an edit disfluency, even though the word sequence is repeated. In repetition detection in the HMM approach, we have to predefine some cue words that are SUs and not edit disfluencies (such as ‘*uhhuh uhhuh*’); whereas the probabilistic Maxent model is able to learn these kinds of cue words and thus is more elegant. Also note that in the heuristic rules, the system SU hypotheses are used when determining the onset of reparandum based on the IP hypotheses. In the Maxent approach, such SU information is generated by the Maxent classifier itself.

The features used in the Maxent model for the SU/IP/NULL detection task are as follows:

- All the features used for SU detection.
- Repetition information. At each word boundary, this feature represents whether there is a repeated word sequence (up to 3 words) that ends at that point, with optional filler words allowed starting from that point.

⁵Even if we build a three way classifier, we only use the edit IP results from this component for edit word detection task.

- **Fragment.** This feature represents whether the word is a fragment. Only in the reference transcription condition can this feature be triggered. In the speech recognition output condition, there is no word fragment information.
- **Filler words.** This feature represents whether there is a pre-defined filler phrase after the word boundary.⁶
- **Prosody posterior probabilities.** A decision tree is trained for the binary classification task, IP, or NULL. The posterior probabilities are represented in a cumulative binning way.

5.1.3. CRF

The CRF approach used for edit word detection finds the whole region of the reparandum. This is similar to named entity recognition [18]. In this approach, each word has an associated tag, representing whether it is an edit word or not. The classes in the CRF edit word detection method are: the beginning of an edit, inside of an edit, each of which has a possible IP associated with it, and outside of an edit. There is a total of 5 states in this model, shown in Table 8. The following is an example of targets for a transcript excerpt:

I I work uh i'm an analyst
 B-E+IP I-E I-E+IP O O O O

%and it got it got real rough
 %O B-E I-E+IP O O O O

According to the annotation guideline and MDE task definition, the IPs are annotated inside complex edit disfluencies, and they are scored for IP detection task. Therefore we include IPs in the target class when using CRF for edit detection in order to find the internal IPs inside the complex edit disfluencies. For example, ‘*I I work*’ in the above example is the reparandum in a complex edit disfluency. The goal is not only to find the whole region, but also the internal IPs (e.g., there is one IP after the first ‘*I*’).

Number	Notation	Meaning	State destinations
0	O	outside edit	O, B-E+IP, B-E
1	B-E+IP	begin edit with an IP	O, I-E+IP, I-E
2	B-E	begin edit	I-E+IP, I-E
3	I-E+IP	inside edit with an IP	O, B-E+IP, I-E+IP, I-E
4	I-E	inside edit	I-E+IP, I-E

Table 8. States used in CRF edit word and edit IP detection.

The CRF model is able to learn the sequence information, i.e., the valid state transitions, from the training data.

⁶This is not from the filler word detection results; rather a list of cue words are used.

All the possible states that a state can go to are also shown in Table 8. Valid state transitions can be guaranteed, i.e. only state 1 or 2 (the beginning of the edit state) can transition to state 3 or 4 (inside a reparandum); whereas, state 0 cannot. An advantage of the CRF method is that it is a probabilistic model, and it provides a more principled way to represent this information than using heuristic rules.

Features used in the CRF method are the N-grams of words and POS tags, and all of the features used by the Maxent IP detection model that are not used for SU detection.

Features used in the CRF method are the same word and POS N-grams as used in SU detection task, as well as the repetition, filler, fragment, and the binned posterior probabilities from the IP prosody model, as used in the Maxent model for IP detection.

5.2. Edit Detection Results

5.2.1. Eval Official Results

	Conditions	Error Rate (%)
CTS	REF	51.49
	STT: IBM+SRI	80.72
	STT: SRI	81.36
BN	REF	43.00
	STT: SuperEARS	89.86
	STT: SRI	91.40

Table 9. Eval Edit word detection submission results for BN and CTS on REF and various STT conditions.

Table 9 shows the edit word detection results for both CTS and BN. A CRF approach is used for CTS edit word detection; whereas, a Maxent approach is used on BN. Interestingly, we observe from the table that the edit word error rate on BN is not worse than CTS for the reference condition, even though the percentage of edit words is much smaller on BN than CTS, which significantly affects the denominator used in the performance measure. This suggests that to some extent edit word detection is a relatively easier task on BN than CTS, which makes sense due to the different speaking style of the two corpora. Also note that an important feature for edit word and editIP detection is the occurrence of word fragments, which is provided in the reference condition but unavailable in the STT condition. The severe performance degradation on the STT condition is due to the unavailability of such word fragment information, as well as word errors in the edit disfluency region.

5.2.2. Effect of Different Approaches

We compare the three methods for edit word and IP detection. Table 10 shows the results on the CTS reference con-

dition. In addition to the official edit word detection results, we also show the result for the edit IP detection, since that is the target class in most modeling approaches. Results in the table show that CRF is better at finding edit words, but worse at IP detection compared to the HMM or Maxent methods. This ties into how the models are trained: the HMM and Maxent are trained to detect IPs, but the heuristic rules used may not find the correct onset for the reparandum; whereas, the CRF is trained to jointly detect the edit words and IPs and thus may not be well trained for IPs using the current features.

Approaches	Edit word	edit IP
HMM	54.33	33.21
Maxent	55.89	34.11
CRF	51.49	36.38

Table 10. Edit word and IP detection using the HMM, Maxent, and CRF approaches on CTS reference condition.

6. IP DETECTION TASK

The IP detection results are obtained from the combined edit word detection and filler detection. Table 11 shows the official results for our eval submission. The poorer IP detection results using SuperEARS STT output on BN is due to the poorer filler word detection results as explained before.

	Conditions	Error Rate (%)
CTS	REF	30.31
	STT: IBM+SRI	60.59
	STT: SRI	61.48
BN	REF	21.42
	STT: SuperEARS	70.39
	STT: SRI	67.91

Table 11. Eval IP detection submission results for BN and CTS, on REF and various STT conditions.

7. CONCLUSIONS AND FUTURE WORK

7.1. Ongoing and Future Work

The significant degradation in performance on MDE tasks when using the STT output (versus the true words) motivates an approach that can integrate information from multiple word hypotheses. See [27] for details on the use of N-best lists and lattices for SU detection.

We plan to investigate whether incorporating more syntactic knowledge helps metadata detection. Ongoing research on this includes using SuperARV tags [28] for SU

and edit word detection. For the filler word detection, we plan to build word dependent prosody models (such as for “like”, “so”) and also investigate other classification algorithms (e.g., Maxent) in addition to the HMM approach. Another future direction is the investigation of direct incorporation of the prosodic features into the Maxent or CRF approaches. A joint model for various metadata events is also worthy of study.

7.2. Summary

We have described various knowledge sources and modeling approaches for automatic detection of metadata events in both conversational and broadcast news speech. Prosodic and textual knowledge sources are utilized for effective MDE. We have explored new modeling techniques in an attempt to address problems in the previous HMM approach, i.e., the Maxent and CRF modeling approaches. These methods outperform our previous HMM on most conditions, and their combination achieves the best performance.

8. ACKNOWLEDGMENTS

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

9. REFERENCES

- [1] D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman, “Measuring the readability of automatic speech-to-text transcripts,” in *Proc. of Eurospeech*, 2003, pp. 1585–1588.
- [2] P. Heeman and J. Allen, “Speech repairs, intonational phrases and discourse markers: Modeling speakers’ utterances in spoken dialogue,” *Computational Linguistics*, 1999.
- [3] J. Kim and P. C. Woodland, “The use of prosody in a combined system for punctuation generation and speech recognition,” in *Proc. of Eurospeech*, 2001, pp. 2757–2760.
- [4] Y. Gotoh and S. Renals, “Sentence boundary detection in broadcast speech transcripts,” in *Proc. of ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000*, 2000, pp. 228–235.

- [5] R. Kompe, *Prosody in Speech Understanding System*, Springer-Verlag, 1996.
- [6] M. Snover, B. Dorr, and R. Schwartz, “A lexically-driven algorithm for disfluency detection,” in *Proc. of HLT/NAACL*, 2004.
- [7] J. Kim, “Automatic detection of sentence boundaries, disfluencies, and conversational fillers in spontaneous speech,” M.S. thesis, University of Washington, 2004.
- [8] M. Johnson and E. Charniak, “A TAG-based noisy channel model of speech repairs,” in *Proc. of ACL*, 2004.
- [9] S. Strassel, *Simple Metadata Annotation Specification V6.2*, Linguistic Data Consortium, 2004.
- [10] S. Strassel, *Simple Metadata Annotation Specification V5.0*, Linguistic Data Consortium, 2004.
- [11] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur, “Prosody-based automatic segmentation of speech into sentences and topics,” *Speech Communication*, pp. 127–154, 2000.
- [12] H. Christensen, Y. Gotoh, and S. Renals, “Punctuation annotation using statistical prosody models,” in *ISCA Workshop on Prosody in Speech Recognition and Understanding*, 2001.
- [13] A. Stolcke and E. Shriberg, “Automatic linguistic segmentation of conversational speech,” in *Proc. of the International Conference of the Spoken Language Processing*, 1996, pp. 1005–1008.
- [14] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, “Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech,” in *Proc. of EMNLP*, 2004.
- [15] S. Chen and R. Rosenfeld, “A Gaussian prior for smoothing maximum entropy models,” Tech. Rep., Carnegie Mellon University, 1999.
- [16] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random field: Probabilistic models for segmenting and labeling sequence data,” in *Proc. of ICML 2001*, 2001, pp. 282–289.
- [17] F. Sha and F. Pereira, “Shallow parsing with conditional random fields,” in *Proc. of HLT-NAACL’03*, 2003.
- [18] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields,” in *Proc. of CoNLL*, 2003.
- [19] A. McCallum, “Mallet: A machine learning for language toolkit,” <http://mallet.cs.umass.edu>, 2002.
- [20] Y. Liu, E. Shriberg, A. Stolcke, and M. Harper, “Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection,” in *Proc. of ICSLP*, 2004.
- [21] T. Brants, “TnT a statistical part-of-speech tagger,” in *Proc. of the 6th Applied NLP Conference*, 2000, pp. 224–231.
- [22] P. F. Brown, V. J. D. Pietra, P. V. DeSouza, J. C. Lai, and R. L. Mercer, “Class-based n-gram models of natural language,” *Computational Linguistics*, pp. 467–479, 1992.
- [23] G. Ngai and R. Florian, “Transformation-based learning in the fast lane,” in *Proc. of NAACL 2001*, June 2001, pp. 40–47.
- [24] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, D. Hillard, M. Ostendorf, and M. Harper, “The ICSI-SRI-UW metadata extraction system,” in *Proc. of ICSLP*, 2004.
- [25] C. Wooters, J. Fung, B. Peskin, and X. Anguera, “Towards robust speaker segmentation: The icsi-sri fall 2004 diarization system,” in *Proc. of RT-04 ERAS Workshop*, 2004.
- [26] Y. Liu, E. Shriberg, and A. Stolcke, “Automatic disfluency identification in conversational speech using multiple knowledge sources,” in *Proc. of Eurospeech*, 2003, pp. 957–960.
- [27] D. Hillard, M. Ostendorf, and A. Stolcke, “Accounting for stt uncertainty in MDE,” in *Proc. of RT-04 EARS Workshop*, 2004.
- [28] W. Wang, *Statistical Parsing and Language Modeling Based on Constraint Dependency Grammar*, Ph.D. thesis, Purdue University, 2003.