Comparing and Combining Generative and Posterior Probability Models: Some Advances in Sentence Boundary Detection in Speech

Yang Liu ICSI and Purdue University yangl@icsi.berkeley.edu
Andreas Stolcke, Elizabeth Shriberg SRI and ICSI stolcke,ees@speech.sri.com
Mary Harper Purdue University harper@ecn.purdue.edu

Abstract

We compare and contrast two different models for detecting sentence-like units in continuous speech. The first approach uses hidden Markov sequence models based on N-grams and maximum likelihood estimation, and employs model interpolation to combine different representations of the data. The second approach models the posterior probabilities of the target classes; it is discriminative and integrates multiple knowledge sources in the maximum entropy (maxent) framework. Both models combine lexical, syntactic, and prosodic information. We develop a technique for integrating pretrained probability models into the maxent framework, and show that this approach can improve on an HMM-based state-of-the-art system for the sentence-boundary detection task. An even more substantial improvement is obtained by combining the posterior probabilities of the two systems.

1 Introduction

Sentence boundary detection is a problem that has received limited attention in the text-based computational linguistics community (Schmid, 2000; Palmer and Hearst, 1994; Reynar and Ratnaparkhi, 1997), but which has recently acquired renewed importance through an effort by the DARPA EARS program (DARPA Information Processing Technology Office, 2003) to improve automatic speech transcription technology. Since standard speech recognizers output an unstructured stream of words, improving transcription means not only that word accuracy must be improved, but also that commonly used structural features such as sentence boundaries need to be recognized. The task is thus fundamentally based on both acoustic and textual (via automatic word recognition) information. From a computational linguistics point of view, sentence units are crucial and assumed in most of the further processing steps that one would want to apply to such output: tagging and parsing, information extraction, and summarization, among others.

Sentence segmentation from speech is a difficult problem. The best systems benchmarked in a recent government-administered evaluation yield error rates between 30% and 50%, depending on the genre of speech processed (measured as the number of missed and inserted sentence boundaries as a percentage of true sentence boundaries). Because of the difficulty of the task, which leaves plenty of room for improvement, its relevance to real-world applications, and the range of potential knowledge sources to be modeled (acoustics and text-based, lower- and higher-level), this is an interesting challenge problem for statistical and computational approaches.

All of the systems participating in the recent DARPA RT-03F Metadata Extraction evaluation (National Institute of Standards and Technology, 2003) were based on a hidden Markov model framework, in which word/tag sequences are modeled by N-gram language models (LMs). Additional features (mostly reflecting speech prosody) are modeled as observation likelihoods attached to the N-gram states of the HMM (Shriberg et al., 2000). The HMM is a generative modeling approach, since it describes a stochastic process with hidden variables (the locations of sentence boundaries) that produces the observable data. The segmentation is inferred by comparing the likelihoods of different boundary hypotheses.

While the HMM approach is computationally efficient and (as described later) provides a convenient way for modularizing the knowledge sources, it 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. The latter is a criterion more closely related to classification error. 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; this in turn would make robust estimation

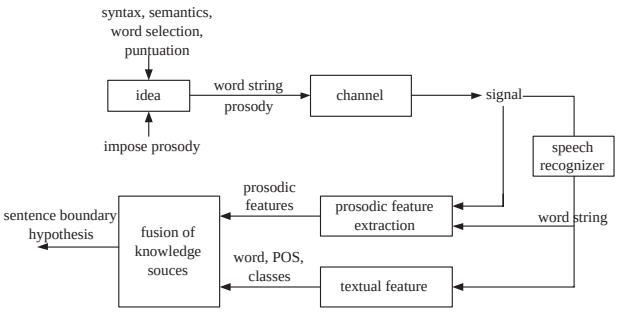

Figure 1: Diagram of the sentence segmentation task.

difficult.

In this paper, we describe our effort to overcome these shortcomings by 1) replacing the generative model with one that estimates the posterior probabilities directly, and 2) using the maximum entropy (maxent) framework to estimate conditional distributions, giving us a more principled way to combine a large number of overlapping features. Both techniques have been used previously for traditional NLP tasks, but they are not straightforward to apply in our case because of the diverse nature of the knowledge sources used in sentence segmentation. We describe the techniques we developed to work around these difficulties, and compare classification accuracy of the old and new approach on different genres of speech. We also investigate how word recognition error affects that comparison. Finally, we show that a simple combination of the two approaches turns out to be highly effective in improving the best previous results obtained on a benchmark task.

2 The Sentence Segmentation Task

The sentence boundary detection problem is depicted in Figure 1 in the source-channel framework. The speaker intends to say something, chooses the word string, and imposes prosodic cues (duration, emphasis, intonation, etc). This signal goes through the speech production channel to generate an acoustic signal. A speech recognizer determines the most likely word string given this signal. To detect possible sentence boundaries in the recognized word string, prosodic features are extracted from the signal, and combined with textual cues obtained from the word string. At issue in this paper is the final box in the diagram: how to model and combine the available knowledge sources to find the most accurate hypotheses.

Note that this problem differs from the sentence boundary detection problem for written text in the natural language processing literature (Schmid, 2000; Palmer and Hearst, 1994; Reynar and Ratnaparkhi, 1997). Here we are dealing with spoken language, therefore there is no punctuation information, the words are not capitalized, and the transcripts from the recognition output are errorful. This lack of textual cues is partly compensated by prosodic information (timing, pitch, and energy patterns) conveyed by speech. Also note that in spontaneous conversational speech "sentence" is not always a straightforward notion. For our purposes we use the definition of a "sentence-like unit", or SU, as defined by the LDC for labeling and evaluation purposes (Strassel, 2003).

The training data has SU boundaries marked by annotators, based on both the recorded speech and its transcription. In testing, a system has to recover both the words and the locations of sentence boundaries, denoted by $(W, E) = w_1 e_1 w_2 \ldots w_i e_i \ldots w_n$ where W represents the strings of word tokens and E the inter-word boundary events (sentence boundary or no boundary).

The system output is scored by first finding a minimum edit distance alignment between the hypothesized word string and the reference, and then comparing the aligned event labels. The SU error rate is defined as the total number of deleted or inserted SU boundary events, divided by the number of true SU boundaries.[1] For diagnostic purposes a secondary evaluation condition allows use of the correct word transcripts. This condition allows us to study the segmentation task without the confounding effect of speech recognition errors, using perfect lexical information.

3 Features and Knowledge Sources

Words and sentence boundaries are mutually constrained via syntactic structure. Therefore, the word identities themselves (from automatic recognition or human transcripts) constitute a primary knowledge source for the sentence segmentation task. We also make use of various automatic taggers that map the word sequence to other representations. The TnT tagger (Brants, 2000) is used to obtain part-of-speech (POS) tags. A TBL chunker trained on Wall Street Journal corpus (Ngai and Florian, 2001) maps each word to an associated chunk tag, encoding chunk type and relative word position (beginning of an NP, inside a VP, etc.). 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 esti-

[1] This is the same as simple per-event classification accuracy, except that the denominator counts only the "marked" events, thereby yielding error rates that are much higher than if one uses all potential boundary locations.

mates. For the same reasons we also generate word class labels that are automatically induced from bigram word distributions (Brown et al., 1992).

To model the prosodic structure of sentence boundaries, we extract several hundred features around each word boundary. These are based on the acoustic alignments produced by a speech recognizer (or forced alignments of the true words when given). The features capture duration, pitch, and energy patterns associated with the word boundaries. Informative features include the pause duration at the boundary, the difference in pitch before and after the boundary, and so on. A crucial aspect of many of these features is that they are highly correlated (e.g., by being derived from the same raw measurements via different normalizations), real-valued (not discrete), and possibly *undefined* (e.g., unvoiced speech regions have no pitch). These properties make prosodic features difficult to model directly in either of the approaches we are examining in the paper. Hence, we have resorted to a modular approach: the information from prosodic features is modeled separately by a decision tree classifier that outputs posterior probability estimates $P(e_i|\boldsymbol{f}_i)$, where e_i is the boundary event after w_i, and $\boldsymbol{f}_i$ is the prosodic feature vector associated with the word boundary. Conveniently, this approach also permits us to include some non-prosodic features that are highly relevant for the task, but not otherwise represented, such as whether a speaker (turn) change occurred at the location in question.[2]

A practical issue that greatly influences model design is that not all information sources are available uniformly for all training data. For example, prosodic modeling assumes acoustic data; whereas, word-based models can be trained on text-only data, which is usually available in much larger quantities. This poses a problem for approaches that model all relevant information jointly and is another strong motivation for modular approaches.

4 The Models

4.1 Hidden Markov Model for Segmentation

Our baseline model, and the one that forms the basis of much of the prior work on acoustic sentence segmentation (Shriberg et al., 2000; Gotoh and Renals, 2000; Christensen, 2001; Kim and Woodland, 2001), is a hidden Markov model. The states of the model correspond to words w_i and following event labels e_i. The observations associated with the states are the words, as well as other (mainly prosodic) features $\boldsymbol{f}_i$. Figure 2 shows a graphical model representation of the variables involved. 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.

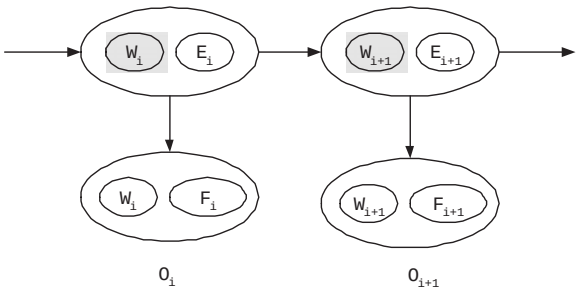

Figure 2: The graphical model for the SU detection problem. Only one word+event is depicted in each state, but in a model based on N-grams the previous $N-1$ tokens would condition the transition to the next state.

4.1.1 Classification

Standard algorithms are available to extract the most probable state (and thus event) sequence given a set of observations. The error metric is based on classification of individual word boundaries. Therefore, rather than finding the highest probability *sequence* of events, we identify the events with highest posterior *individually* at each boundary i:

$$\hat{e}_i = \arg\max_{e_i} P(e_i|W, F) \quad (1)$$

where W and F are the words and features for the entire test sequence, respectively. The individual event posteriors are obtained by applying the forward-backward algorithm for HMMs (Rabiner and Juang, 1986).

4.1.2 Model Estimation

Training of the HMM is supervised since event-labeled data is available. There are two sets of parameters to estimate. The state transition probabilities are estimated using a hidden event N-gram LM (Stolcke and Shriberg, 1996). The LM is obtained with standard N-gram estimation methods from data that contains the word+event tags in sequence: $w_1, e_1, w_2, \ldots e_{n-1}, w_n$. The resulting LM can then compute the required HMM transition

[2]Here we are glossing over some details on prosodic modeling that are orthogonal to the discussion in this paper. For example, instead of simple decision trees we actually use ensemble bagging to reduce the variance of the classifier (Liu et al., 2004).

probabilities as[3]

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

The N-gram estimator maximizes the joint word+event sequence likelihood $P(W, E)$ on the training data (modulo smoothing), and does not guarantee that the correct event posteriors needed for classification according to Equation (1) are maximized.

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 described in Section 3. We have at our disposal decision trees that estimate $P(e_i|f_i)$. If we further assume that prosodic features are independent of words given the event type (a reasonable simplification if features are chosen appropriately), observation likelihoods may be obtained by

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

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

4.1.3 Knowledge Combination

The HMM structure makes strong independence assumptions: (1) that features depend only on the current state (and in practice, as we saw, only on the event label) and (2) that each word+event label depends only on the last $N - 1$ tokens. In return, we get a computationally efficient structure that allows information from the entire sequence W, F to inform the posterior probabilities needed for classification, via the forward-backward algorithm.

More problematic in practice is the integration of multiple word-level features, such as POS tags and chunker output. Theoretically, all tags could simply be included in the hidden state representation to allow joint modeling of words, tags, and events. However, this would drastically increase the size of the state space, making robust model estimation with standard N-gram techniques difficult. A method that works well in practice is *linear interpolation*, whereby the conditional probability estimates of various models are simply averaged, thus reducing variance. In our case, we obtain good results by interpolating a word-N-gram model with one based on automatically induced word classes (Brown et al., 1992).

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 training data labeled precisely to the LDC's SU specifications, but a much larger (130M word) corpus of standard broadcast new 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.

4.2 Maxent Posterior Probability Model

As observed, HMM training does not maximize the posterior probabilities of the correct labels. This mismatch between training and use of the model as a classifier would not arise if the model directly estimated the posterior boundary label probabilities $P(e_i|W, F)$. A second problem with HMMs is that the underlying N-gram sequence model does not cope well with multiple representations (features) of the word sequence (words, POS, etc.) short of building a joint model of all variables. This type of situation is well-suited to a maximum entropy formulation (Berger et al., 1996), which allows conditioning features to apply simultaneously, and therefore gives greater freedom in choosing representations. Another desirable characteristic of maxent models is that they do not split the data recursively to condition their probability estimates, which makes them more robust than decision trees when training data is limited.

4.2.1 Model Formulation and Training

We built a posterior probability model for sentence boundary classification in the maxent framework. Such a model takes the familiar exponential form[4]

$$P(e|W, F) = \frac{1}{Z_\lambda(W, F)} e^{\sum_k \lambda_k g_k(e, W, F)} \qquad (3)$$

where $Z_\lambda(W, F)$ is the normalization term:

$$Z_\lambda(W, F) = \sum_{e'} e^{\sum_k \lambda_k g_k(e', W, F)} \qquad (4)$$

The functions $g_k(e, W, F)$ are indicator functions corresponding to (complex) features defined over

[3] To 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.

[4] We omit the index i from e here since the "current" event is meant in all cases.

events, words, and prosodic features. For example, one such feature function might be:

$$g(e, W, F) = \begin{cases} 1 : \text{if } w_i = \textit{uhhuh} \text{ and } e = SU \\ 0 : \text{otherwise} \end{cases}$$

The maxent model is estimated by finding the parameters λ_k such that the expected values of the various feature functions $E_P[g_k(e', W, F)]$ match the empirical averages in the training data. It can be shown that the resulting model has maximal entropy among all the distributions satisfying these expectation constraints. At the same time, the parameters so chosen maximize the conditional likelihood $\prod_i P(e_i|W, F)$ over the training data, subject to the constraints of the exponential form given by Equation (3).[5] 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.

4.2.2 Choice of Features

Even though the mathematical formulation gives us the freedom to use features that are overlapping or otherwise dependent, we still have to choose a subset that is informative and parsimonious, so as to give good generalization and robust parameter estimates. Various feature selection algorithms for maxent models have been proposed, e.g., (Berger et al., 1996). However, since computational efficiency was not an issue in our experiments, we included all features that corresponded to information available to our baseline approach, as listed below. We did eliminate features that were triggered only once in the training set to improve robustness and to avoid overconstraining the model.

- *Word N-grams.* We use combinations of preceding and following words to encode the word context of the event, 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 $<w_i, w_{i+1}, w_{i+2}>$, where w_i refers to the word before the boundary of interest.
- *POS N-grams.* POS tags are the same as used for the HMM approach. The features capturing POS context are similar to those based on word tokens.
- *Chunker tags.* These are used similarly to POS and word features, except we use tags encoding chunk type (NP, VP, etc.) and word position within the chunk (beginning versus inside).[6]
- *Word classes.* These are similar to N-gram patterns but over automatically induced classes.
- *Turn flags.* Since speaker change often marks an SU boundary, we use this binary feature. Note that in the HMM approach this feature had to be grouped with the prosodic features and handled by the decision tree. In the maxent approach we can use it separately.
- *Prosody.* As we described earlier, decision tree classifiers are used to generate the posterior probabilities $p(e_i|f_i)$. Since the maxent classifier is most conveniently used with binary features, we encode the prosodic posteriors into several binary features via thresholding. Equation (3) 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 $e^{\lambda_k g_k}$). This suggests encoding the decision tree posteriors in a cumulative fashion through a series of binary features, for example, $p > 0.1$, $p > 0.3$, $p > 0.5$, $p > 0.7$, $p > 0.9$. This representation is also more robust to mismatch between the posterior probability in training and test set, since small changes in the posterior value affect at most one feature.

  Note that the maxent framework does allow the use of real-valued feature functions, but preliminary experiments have shown no gain compared to the binary features constructed as described above. Still, this is a topic for future research.
- *Auxiliary LM.* As mentioned earlier, additional text-only language model training data is often available. In the HMM model we incorporated auxiliary LMs by interpolation, which is not possible here since there is no LM *per se*, but rather N-gram features. However, we can use the same trick as we used for prosodic features. A word-only HMM is used to estimate posterior event probabilities according to the auxiliary LM, and these posteriors are then thresholded to yield binary features.
- *Combined features.* To date we have not fully investigated compound features that combine different knowledge sources and are able to model the interaction between them explicitly.

[5] In our experiments we used the L-BFGS parameter estimation method, with Gaussian-prior smoothing (Chen and Rosenfeld, 1999) to avoid overfitting.

[6] Chunker features were only used for broadcast news data, due to the poor chunking performance on conversational speech.

We only included a limited set of such features, for example, a combination of the decision tree hypothesis and POS contexts.

4.3 Differences Between HMM and Maxent

We have already discussed the differences between the two approaches regarding the training objective function (joint likelihood versus conditional likelihood) and with respect to the handling of dependent word features (model interpolation versus integrated modeling via maxent). On both counts the maxent classifier should be superior to the HMM. However, the maxent approach also has some theoretical disadvantages compared to the HMM by design. One obvious shortcoming is that some information gets lost in the thresholding that converts posterior probabilities from the prosodic model and the auxiliary LM into binary features.

A more qualitative limitation of the maxent model is that it only uses local evidence (the surrounding word context and the local prosodic features). In that respect, the maxent model resembles the conditional probability model at the individual HMM states. The HMM as a whole, however, through the forward-backward procedure, propagates evidence from all parts of the observation sequence to any given decision point. Variants such as the conditional Markov model (CMM) combine sequence modeling with posterior probability (e.g., maxent) modeling, but it has been shown that CMM's are still structurally inferior to HMMs because they only propagate evidence forward in time, not backwards (Klein and Manning, 2002).

5 Results and Discussion

5.1 Experimental Setup

Experiments comparing the two modeling approaches were conducted on two corpora: broadcast news (BN) and conversational telephone speech (CTS). BN and CTS differ in genre and speaking style. These differences are reflected in the frequency of SU boundaries: about 14% of inter-word boundaries are SUs in CTS, compared to roughly 8% in BN.

The corpora are annotated by LDC according to the guidelines of (Strassel, 2003). Training and test data are those used in the DARPA Rich Transcription Fall 2003 evaluation.[7] For CTS, there is about 40 hours of conversational data from the Switchboard corpus for training and 6 hours (72 conversations) for testing. The BN data has about 20 hours of broadcast news shows in the training set and 3 hours (6 shows) in the test set. The SU detection task is evaluated on both the reference transcriptions (REF) and speech recognition outputs (STT). The speech recognition output is obtained from the SRI recognizer (Stolcke et al., 2003).

System performance is evaluated using the official NIST evaluation tools,[8] which implement the metric described earlier. In our experiments, we compare how the two approaches perform individually and in combination. The combined classifier is obtained by simply averaging the posterior estimates from the two models, and then picking the event type with the highest probability at each position.

We also investigate other experimental factors, such as the impact of the speech recognition errors, the impact of genre, and the contribution of text versus prosodic information in each model.

		HMM	Maxent	Combined
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 1: SU detection results (error rate in %) using maxent and HMM individually and in combination on BN and CTS.

5.2 Experimental Results

Table 1 shows SU detection results for BN and CTS, using both reference transcriptions and speech recognition output, using the HMM and the maxent approach individually and in combination. The maxent approach slightly outperforms the HMM approach when evaluating on the reference transcripts, and the combination of the two approaches achieves the best performance for all tasks (significant at $p < 0.05$ using the sign test on the reference transcription condition, mixed results on using recognition output).

5.2.1 BN vs. CTS

The detection error rate on CTS is lower than on BN. This may be due to the metric used for performance. Detection error rate is measured as the percentage of errors per reference SU. The number of SUs in CTS is much larger than for BN, making the relative error rate lower for the conversational speech task. Notice also from Table 1 that maxent yields more gain on CTS than on BN (for the reference transcription condition on both corpora). One possible reason for this is that we have more train-

[7] We used both the development set and the evaluation set as the test set in this paper, in order to have a larger test set to make the results more meaningful.

[8] http://www.nist.gov/speech/tests/rt/rt2003/fall/

		Del	Ins	Total
BN	HMM	28.48	20.24	48.72
	Maxent	32.06	16.54	48.61
CTS	HMM	17.19	14.32	31.51
	Maxent	19.97	10.69	30.66

Table 2: Error rates for the two approaches on reference transcriptions. Performance is shown in deletion, insertion, and total error rate (%).

		BN	CTS
HMM	Textual	67.48	38.92
	Textual + prosody	48.72	31.51
Maxent	Textual	63.56	36.32
	Textual + prosody	48.61	30.66

Table 3: SU detection error rate (%) using different knowledge sources, for BN and CTS, evaluated on the reference transcription.

ing data and thus less of a sparse data problem for CTS.

5.2.2 Error Type Analysis

Table 2 shows error rates for the HMM and the maxent approaches in the reference condition. Due to the reduced dependence on the prosody model, the errors made in the maxent approach are different from the HMM approach. There are more deletion errors and fewer insertion errors, since the prosody model tends to overgenerate SU hypotheses. The different error patterns suggest that we can effectively combine the system output from the two approaches. As shown in the Table 1, the combination of maxent and HMM consistently yields the best performance.

5.2.3 Contribution of Knowledge Sources

Table 3 shows SU detection results for the two approaches, using textual information only, as well as in combination with the prosody model (which are the same results as shown in Table 1). We only report the results on the reference transcription condition, in order to not confound the comparison by word recognition errors.

The superior results for text-only classification are consistent with the maxent model's ability to combine overlapping word-level features in a principled way. However, the HMM largely catches up once prosodic information is added. This can be attributed to the loss-less integration of prosodic posteriors in the HMM, as well as the fact that in the HMM, each boundary decision is affected by prosodic information throughout the data; whereas, the maxent model only uses the prosodic features at the boundary to be classified.

5.2.4 Effect of Recognition Errors

We observe in Table 1 that there is a large increase in error rate when evaluating on the speech recognition output. This happens in part because word information is inaccurate in the recognition output, thus impacting the LMs and lexical features. The prosody model is also affected, since the alignment of incorrect words to the speech is imperfect, thereby affecting the prosodic feature extraction. However, the prosody model is more robust to recognition errors than the LMs, due to its lesser dependence on word identity. The degradation on CTS is larger than on BN. This can easily be explained by the difference in word error rates, 22.9% on CTS and 12.1% on BN.

The maxent system degrades more than then HMM system when errorful recognition output is used. In light of the previous section, this makes sense: most of the improvement of the maxent model comes from better lexical feature modeling. But these are exactly the features that are most deteriorated by faulty recognition output.

6 Conclusions and Future Work

We have described two different approaches for modeling and integration of diverse knowledge sources for automatic sentence segmentation from speech: a state-of-the-art approach based on HMMs, and an alternative approach based on posterior probability estimation via maximum entropy. To achieve competitive performance with the maxent model we devised a cumulative binary coding scheme to map posterior estimates from auxiliary submodels into features for the maxent model.

The two approaches have complementary strengths and weaknesses that were reflected in the results, consistent with the findings for text-based NLP tasks (Klein and Manning, 2002). The maxent model showed much better accuracy than the HMM with lexical information, and a smaller win after combination with prosodic features. The HMM made more effective use of prosodic information and degraded less with errorful word recognition. A interpolation of posterior probabilities from the two systems achieved 2-7% relative error reduction compared to the baseline (significant at $p < 0.05$ for the reference transcription condition). The results were consistent for two different genres of speech.

In future work we hope to determine how the individual qualitative differences of the two models (estimation methods, model structure, etc.) contribute to the observed differences in results. To improve results overall, we plan to explore features

that combine multiple knowledge sources, as well as approaches that model recognition uncertainty in order to mitigate the effects of word errors. We also plan to investigate using a conditional random field (CRF) models. CRFs combine the advantages of both the HMM and the maxent approaches, being a discriminatively trained model that can incorporate overlapping features (the maxent advantages), while also modeling sequence dependencies (an advantage of HMMs) (Lafferty et al., 2001).

7 Acknowledgments

The authors gratefully thank Le Zhang for his guidance in applying the maximum entropy approach to this task. This research has been supported by DARPA under contract MDA972-02-C-0038, NSF-STIMULATE under IRI-9619921, NSF BCS-9980054, 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, or NASA. Part of this work was carried out while the last author was on leave from Purdue University and at NSF.

References

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–72.

T. Brants. 2000. TnT—a statistical part-of-speech tagger. In *Proc. of the Sixth Applied NLP*, pages 224–231.

P. F. Brown, V. J. Della Pietra, P. V. DeSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

S. Chen and R. Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

H. Christensen. 2001. Punctuation annotation using statistical prosody models. In *ISCA Workshop on Prosody in Speech Recognition and Understanding*.

DARPA Information Processing Technology Office. 2003. Effective, affordable, reusable speech-to-text (EARS). http://www.darpa.mil/ipto/programs/ears/.

Y. Gotoh and S. Renals. 2000. Sentence boundary detection in broadcast speech transcripts. In *ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000*, pages 228–235.

J. Kim and P. C. Woodland. 2001. The use of prosody in a combined system for punctuation generation and speech recognition. In *Proc. of Eurospeech 2001*, pages 2757–2760.

D. Klein and C. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proc. of EMNLP 2002*, pages 9–16.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random field: Probabilistic models for segmenting and labeling sequence data. In *Prof. of ICML 2001*, pages 282–289.

Y. Liu, E. Shriberg, A. Stolcke, and M. Harper. 2004. Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection. In *Proc. of ICSLP 2004 (To Appear)*.

National Institute of Standards and Technology. 2003. RT-03F workshop agenda and presentations. http://www.nist.gov/speech/tests/rt/rt2003/fall/presentations/, November.

G. Ngai and R. Florian. 2001. Transformation-based learning in the fast lane. In *Proc. of NAACL 2001*, pages 40–47, June.

D. D. Palmer and M. A. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proc. of the Fourth Applied NLP*, pages 78–83.

L. R. Rabiner and B. H. Juang. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, January.

J. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proc. of the Fifth Applied NLP*, pages 16–19.

H. Schmid. 2000. Unsupervised learning of period disambiguation for tokenisation. University of Stuttgart, Internal Report.

E. Shriberg, A. Stolcke, D. H. Tur, and G. Tur. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154.

A. Stolcke and E. Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Proc. of ICSLP 1996*, pages 1005–1008.

A. Stolcke, H. Franco, and R. Gadde et al. 2003. Speech-to-text research at SRI-ICSI-UW. http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/index.htm.

S. Strassel, 2003. *Simple Metadata Annotation Specification V5.0*. Linguistic Data Consortium.