Effective Use of Prosody in Parsing Conversational Speech

Jeremy G. Kahn[†] **Matthew Lease**[*]
Eugene Charniak[*] **Mark Johnson**[*] **Mari Ostendorf**[†]

University of Washington, SSLI[†] Brown University[*]
{jgk,mo}@ssli.ee.washington.edu {mlease,ec,mj}@cs.brown.edu

Abstract

We identify a set of prosodic cues for parsing conversational speech and show how such features can be effectively incorporated into a statistical parsing model. On the Switchboard corpus of conversational speech, the system achieves improved parse accuracy over a state-of-the-art system which uses only lexical and syntactic features. Since removal of edit regions is known to improve downstream parse accuracy, we explore alternatives for edit detection and show that PCFGs are not competitive with more specialized techniques.

1 Introduction

For more than a decade, the Penn Treebank's Wall Street Journal corpus has served as a benchmark for developing and evaluating statistical parsing techniques (Collins, 2000; Charniak and Johnson, 2005). While this common benchmark has served as a valuable shared task for focusing community effort, it has unfortunately led to the relative neglect of other genres, particularly speech. Parsed speech stands to benefit from practically every application envisioned for parsed text, including machine translation, information extraction, and language modeling. In contrast to text, however, speech (in particular, conversational speech) presents a distinct set of opportunities and challenges. While new obstacles arise from the presence of speech repairs, the possibility of word errors, and the absence of punctuation and sentence boundaries, speech also presents a tremendous opportunity to leverage multi-modal input, in the form of acoustic or even visual cues.

As a step in this direction, this paper identifies a set of useful prosodic features and describes how they can be effectively incorporated into a statistical parsing model, ignoring for now the problem of word errors. Evaluated on the Switchboard corpus of conversational telephone speech (Graff and Bird, 2000), our prosody-aware parser out-performs a state-of-the-art system that uses lexical and syntactic features only. While we are not the first to employ prosodic cues in a statistical parsing model, previous efforts (Gregory et al., 2004; Kahn et al., 2004) incorporated these features as word tokens and thereby suffered from the side-effect of displacing words in the n-gram models by the parser. To avoid this problem, we generate a set of candidate parses using an off-the-shelf, k-best parser, and use prosodic (and other) features to rescore the candidate parses.

Our system architecture combines earlier models proposed for parse reranking (Collins, 2000) and filtering out edit regions (Charniak and Johnson, 2001). Detecting and removing edits prior to parsing is motivated by the claim that probabilistic context-free grammars (PCFGs) perform poorly at detecting edit regions. We validate this claim empirically: two state-of-the-art PCFGs (Bikel, 2004; Charniak and Johnson, 2005) are both shown to perform significantly below a state-of-the-art edit detection system (Johnson et al., 2004).

2 Previous Work

As mentioned earlier, conversational speech presents a different set of challenges and opportunities than encountered in parsing text. This paper focuses on the challenges associated with disfluencies (Sec. 2.1) and the opportunity of leveraging acoustic-prosodic cues at the sub-sentence level (Sec. 2.2). Here, sentence segmentation is assumed to be known (though punctuation is not available);

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 233–240, Vancouver, October 2005.

$$\text{...while}\quad \underbrace{\text{I think,}}_{\text{Reparandum}}\quad + \quad \underbrace{\text{uh, I mean,}}_{\text{Editing phrase}}\ \underbrace{\text{I know}}_{\text{Repair}}\ \text{that...}$$

Figure 1: The structure of a typical repair, with "+" indicating the interruption point.

the impact of automatic segmentation is addressed in other work (Kahn et al., 2004).

2.1 Speech Repairs and Parsing

Spontaneous speech abounds with disfluencies such as partial words, filled pauses (e.g., "uh", "um"), conversational fillers (e.g., "you know"), and parenthetical asides. One type of disfluency that has proven particularly problematic for parsing is speech repairs: when a speaker amends what he is saying mid-sentence (see Figure 1). Following the analysis of (Shriberg, 1994), a speech repair can be understood as consisting of three parts: the *reparandum* (the material repaired), the *editing phrase* (that is typically either empty or consists of a filler), and the *repair*. The point between the reparandum and the editing phrase is referred to as the *interruption point* (IP), and it is the point that may be acoustically marked. We refer to the reparandum and editing phrase together as an *edit* or *edit region*. Speech repairs are difficult to model with HMM or PCFG models, because these models can induce only linear or tree-structured dependencies between words. The relationship between reparandum and repair is quite different: the repair is often a "rough copy" of the reparandum, using the same or very similar words in roughly the same order. A language model characterizing this dependency with hidden stack operations is proposed in (Heeman and Allen, 1999).

Several parsing models have been proposed which accord special treatment to speech repairs. Most prior work has focused on handling disfluencies and continued to rely on hand-annotated transcripts that include punctuation, case, and known sentence boundaries (Hindle, 1983; Core and Schubert, 1999; Charniak and Johnson, 2001; Engel et al., 2002).

Of particular mention is the analysis of the relationship between speech repairs and parsing accuracy presented by Charniak and Johnson (2001), as this directly influenced our work. They presented evidence that improved edit detection (i.e. detecting the reparandum and edit phrase) leads to better parsing accuracy, showing a relative reduction in F-score error of 14% (2% absolute) between oracle and automatic edit removal. Thus, this work adopts their edit detection preprocessing approach. They have subsequently presented an improved model for detecting edits (Johnson et al., 2004), and our results here complement their analysis of the edit detection and parsing relationship, particularly with respect to the limitations of PCFGs in edit detection.

2.2 Prosody and parsing

While spontaneous speech poses problems for parsing due to the presence of disfluencies and lack of punctuation, there is information in speech associated with prosodic cues that can be taken advantage of in parsing. Certainly, prosodic cues are useful for sentence segmentation (Liu *et al.*, 2004), and the quality of automatic segmentation can have a significant impact on parser performance (Kahn et al., 2004). There is also perceptual evidence that prosody provides cues to human listeners that aid in syntactic disambiguation (Price *et al.*, 1991), and the most important of these cues seems to be the prosodic phrases (perceived groupings of words) or the boundary events marking them. However, the utility of sentence-internal prosody in parsing conversational speech is not well established.

Most early work on integrating prosody in parsing was in the context of human-computer dialog systems, where parsers typically operated on isolated utterances. The primary use of prosody was to rule out candidate parses (Bear and Price, 1990; Batliner *et al.*, 1996). Since then, parsing has advanced considerably, and the use of statistical parsers makes the candidate pruning benefits of prosody less important. This raises the question of whether prosody is useful for improving parsing accuracy for conversational speech, apart from its use in sentence

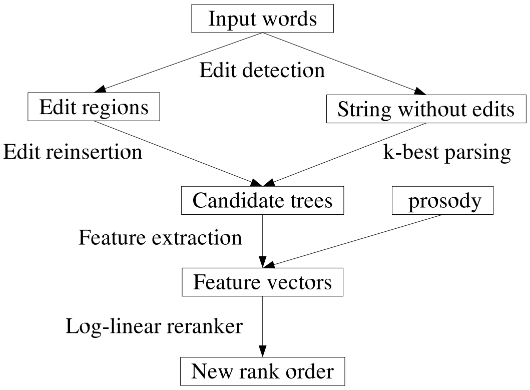

Figure 2: System architecture

boundary detection. Extensions of Charniak and Johnson (2001) look at using quantized combinations of prosodic features as additional "words", similar to the use of punctuation in parsing written text (Gregory et al., 2004), but do not find that the prosodic features are useful. It may be that with the short "sentences" in spontaneous speech, sentence-internal prosody is rarely of use in parsing. However, in edit detection using a parsing model (Johnson et al., 2004), posterior probabilities of automatically detected IPs based on prosodic cues (Liu *et al.*, 2004) are found to be useful. The seeming discrepancy between results could be explained if prosodic cues to IPs are useful but not other sub-sentence prosodic constituents. Alternatively, it could be that including a representation of prosodic features as terminals in (Gregory et al., 2004) displaces words in the parser n-gram model history. Here, prosodic event posteriors are used, with the goal of providing a more effective way of incorporating prosody than a word-like representation.

3 Approach

3.1 Overall architecture

Our architecture, shown in Figure 2, combines the parse reranking framework of (Collins, 2000) with the edit detection and parsing approach of (Charniak and Johnson, 2001). The system operates as follows:

1. Edit words are identified and removed.
2. Each resulting string is parsed to produce a set of k candidate parses.
3. Edit words reinserted into the candidates with a new part-of-speech tag EW. Consecutive sequences of edit words are inserted as single, flat EDITED constituents.
4. Features (syntactic and/or prosodic) are extracted for each candidate, i.e. candidates are converted to feature vector representation.
5. The candidates are rescored by the reranker to identify the best parse.

Use of Collins' parse reranking model has several advantages for our work. In addition to allowing us to incorporate prosody without blocking lexical dependencies, the discriminative model makes it relatively easy to experiment with a variety of prosodic features, something which is considerably more difficult to do directly with a typical PCFG parser.

Our use of the Charniak-Johnson approach of separately detecting disfluencies is motivated by their result that edit detection error degrades parser accuracy, but we also include experiments that omit this step (forcing the PCFG to model the edits) and confirm the practical benefit of separating responsibilities between the edit detection and parsing tasks.

3.2 Baseline system

We adopt an existing parser-reranker as our baseline (Charniak and Johnson, 2005). The parser component supports k-best parse generation, and the reranker component is used to rescore candidate parses proposed by the parser. In detail, the reranker selects from the set of k candidates $T = \{t_1, \ldots t_k\}$ the parse $t^\star \in T$ with the highest bracket F-score (in comparison with a hand-annotated reference). To accomplish this, a feature-extractor converts each candidate parse $t \in T$ into a vector of real-valued features $f(t) = (f_1(t), \ldots, f_m(t))$ (e.g., the value $f_j(t)$ of the feature f_j might be the number of times a certain syntactic structure appears in t). The reranker training procedure associates each feature f_j with a real-valued weight λ_j, and $\lambda' f(t)$ (the dot product of the feature vector and the weight vector λ) is a single scalar weight for each parse candidate. The reranker employs a maximum-entropy estimator that selects the λ that minimizes the log loss of the highest bracket F-score parse $t^\star$ conditioned on T (together with a Gaussian regularizer to prevent overtraining). Informally, λ is chosen to make high F-score parses as likely as possible under the (conditional) distribution defined by f and λ. As in (Collins, 2000), we generate training data for the reranker by reparsing the training corpus, using $n-1$ folds as training data to parse the n-th fold.

The existing system also includes a feature extractor that identifies interesting syntactic relationships not included in the PCFG parsing model (but used in the reranker). These features are primarily related to non-local dependencies, including parallelism of conjunctions, the number of terminals dominated by coordinated structures, right-branching root-to-leaf length, lexical/functional head pairs, n-gram style sibling relationships, etc.

3.3 Prosodic Features

Most theories of prosody have a symbolic representation for prosodic phrasing, where different combinations of acoustic cues (fundamental frequency, energy, timing) combine to give categorical perceptual differences. Our approach to integrating prosody in parsing is to use such symbolic boundary events, including prosodic break labels that build on linguistic notions of intonational phrases and hesitation phenomena. These events are predicted from a combination of continuous acoustic correlates, rather than using the acoustic features directly, because the intermediate representation simplifies training with high-level (sparse) structures. Just as phone-based acoustic models are useful in speech recognition systems as an intermediate level between words and acoustic features (especially for characterizing unseen words), the small set of prosodic boundary events are used here to simplify modeling the inter-dependent set of continuous-valued acoustic cues related to prosody. However, also as in speech recognition, we use posterior probabilities of these events as features rather than making hard decisions about presence vs. absence of a constituent boundary.

In the past, the idea of using perceptual categories has been dismissed as impractical due to the high cost of hand annotation. However, with advances in weakly supervised learning, it is possible to train prosodic event classifiers with only a small amount of hand-labeled data by leveraging information in syntactic parses of unlabeled data. Our strategy is similar to that proposed in (Nöth *et al.*, 2000), which uses categorical labels defined in terms of syntactic structure and pause duration. However, their system's category definitions are without reference to human perception, while we leverage learned relations between perceptual events and syntax with other acoustic cues, without predetermining the relation or requiring a direct coupling to syntax.

More specifically, we represent three classes of prosodic boundaries (or, breaks): major intonational phrase, hesitation, and all other word boundaries.[1] A small set of hand-labeled data from the treebanked portion of the Switchboard corpus (Ostendorf *et al.*, 2001) was used to train initial break prediction models based on both parse and acoustic cues. Next, the full set of treebanked Switchboard data is used with an EM algorithm that iterates between: i) finding probabilities of prosodic breaks in unlabeled data based on the current model, again using parse and acoustic features, and ii) re-estimating the model using the probabilities as weighted counts. Finally, a new acoustic-only break prediction model was designed from this larger data set for use in the parsing experiments.

In each stage, we use decision trees for models, in part because of an interest in analyzing the prosody-syntax relationships learned. The baseline system trained on hand-labeled data has error rates of 9.6% when all available cues are used (both syntax and prosody) and 16.7% when just acoustic and part-of-speech cues are provided (our target environment). Using weakly supervised (EM) training to incorporate unannotated data led to a 15% reduction in error rate to 14.2% for the target trees. The final decision tree was used to generate posteriors for each of the three classes, one for each word in a sentence.

¿From perceptual studies and decision tree analyses, we know that major prosodic breaks tend to co-occur with major clauses, and that hesitations often occur in edit regions or at high perplexity points in the word sequence. To represent the co-occurrence as a feature for use in parse reranking, we treat the prosodic break posteriors as weighted counts in accumulating the number of constituents in parse t of type i with break type j at their right edge, which (with some normalization and binning) becomes feature f_{ij}. Note that the unweighted count

[1] The intonational phrase corresponds to a break of "4" in the ToBI labeling system (Pitrelli et al., 1994), and a hesitation is marked with the "p" diacritic.

for constituent i corresponds directly to a feature in the baseline set, but the baseline set of features also includes semantic information via association with specific words. Here, we simply use syntactic constituents. It is also known that major prosodic breaks tend to be associated with longer syntactic constituents, so we used the weighted count strategy with length-related features as well. In all, the various attributes associated with prosodic break counts were the constituent label of the subtree, its length (in words), its height (maximal distance from the constituent root to any leaf), and the depth of the rightmost word (distance from the right word to the subtree root). For each type in each of these categories, there are three prosodic features, corresponding to the three break types.

3.4 Edit detection

To provide a competitive baseline for our parsing experiments, we used an off-the-shelf, state-of-the-art TAG-based model as our primary edit detector (Johnson et al., 2004).[2] This also provided us a competitive benchmark for contrasting the accuracy of PCFGs on the edit detection task (Section 4.2).

Whereas the crossing-dependencies inherent in speech repairs makes them difficult to model using HMM or PCFG approaches (Section 2.1), Tree Adjoining Grammars (TAGs) are capable of capturing these dependencies. To model both the crossed-dependencies of speech repairs and the linear or tree-structured dependencies of non-repaired speech, Johnson et al.'s system applies the noisy channel paradigm: a PCFG language model defines a probability distribution over non-repaired speech, and a TAG is used to model the optional insertion of edits. The output of this noisy channel model is a set of candidate edits which are then reranked using a max-ent model (similar to what is done here for parse reranking). This reranking step enables incorporation of features based on an earlier word-based classifier (Charniak and Johnson, 2001) in addition to output features of the TAG model. Acoustic features are not yet incorporated.

4 Experimental design

4.1 Corpus

Experiments were carried out on conversational speech using the hand-annotated transcripts associated with the Switchboard treebank (Graff and Bird, 2000). As was done in (Kahn et al., 2004), we resegmented the treebank's sentences into V5-style sentence-like units (SUs) (LDC, 2004), since our ultimate goal was to be able to parse speech given automatically detected boundaries. Unfortunately, the original transcription effort did not provide punctuation guidelines, and the Switchboard treebanking was performed on the transcript unchanged, without reference to the audio. As a result, the sentence boundaries sometimes do not match human listener decisions using SU annotation guidelines, with differences mainly corresponding to treatment of discourse markers and backchannels. In the years since the original Switchboard annotation was performed, LDC has iteratively refined guidelines for annotating SUs, and significant progress has been made in automatically recovering SU boundaries annotated according to this standard (Liu *et al.*, 2004). To eventually leverage this work, we have taken the Meteer-annotated SUs (Meteer et al., 1995), for which there exists treebanked training data, and automatically adjusted them to be more like the V5 LDC standard, and resegmented the Switchboard treebank accordingly. In cases where the original syntactic constituents span multiple SUs, we discard any constituents violating the SU boundary, and in the event that an SU spans a treebank sentence boundary, a new top-level constituent SUGROUP is inserted to produce a proper tree (and evaluated like any other constituent in the gold tree).[3] While this SU resegmentation makes it difficult to compare our experimental results to past work, we believe this is a necessary step towards developing a more realistic baseline for fully automated parsing of speech.

In addition to resegmention, we removed all punctuation and case from the corpus to more closely reflect the form of output available from a speech recognizer. We retained partial words for consis-

[2] We also evaluated another state-of-the-art edit detection system (Liu *et al.*, 2004) but found that it suffered from a mismatch between the current LDC specification of edits (LDC, 2004) and that used in the treebank.

[3] SU and treebank segments disagree at about 5% in each direction, due mostly to the analysis of discourse markers as conjunctions (sentences of >1 SU) and the separation of backchannels into separate treebank sentences (SUs of >1 sentence).

Table 1: Statistics on the Switchboard division used.

Section	Sides	SUs	Words
Train	1,031	87,599	659,437
Tune	126	13,147	103,500
Test	128	8,726	61,313
Total	1,285	109,472	824,250

Table 2: Edit word detection performance for two word-based PCFGs and the TAG-based edit detector. F-score and error are word-based measures.

Edit Detector	Edit F-score	Edit Error
Bikel-Collins PCFG	65.3	62.1
Charniak PCFG	65.8	59.9
TAG-based	78.2	42.2

Table 3: Parsing F-score of various feature and edit-detector combinations.

	PCFG	TAG	Oracle
Edit F (Table 2)	65.8	78.2	100.0
Parser 1-best	84.4	85.0	86.9
Prosodic feats	85.0	85.6	87.6
Syntactic feats	85.9	86.4	88.4
Combined feats	86.0	86.6	88.6
Oracle-rate	92.6	93.2	95.2

tency with other work (Liu *et al.*, 2004; Johnson et al., 2004), although word fragments would not typically be available from ASR. Finally, of the 1300 total conversation sides, we discarded 15 for which we did not have prosodic data. Our corpus division statistics are given in Table 1. During development, experiments were carried out on the *tune* section; the *test* section was reserved for a final test run.

4.2 Experimental Variables

Our primary goal is to evaluate the extent to which prosodic cues could augment and/or stand-in for lexical and syntactic features. Correspondingly, we report on using three flavors of *feature extraction*: syntactic and lexical features (Section 3.2), prosodic features (Section 3.3), and both sets of features combined. For all three conditions, the first-stage score for each parse (generated by the off-the-shelf k-best parser) was always included as a feature.

A second parameter varied in the experiments was the method of upstream *edit detection* employed prior to parsing: PCFG, TAG-based, and oracle knowledge of treebank edit annotations. While it had been claimed that PCFGs perform poorly as edit detectors (Charniak and Johnson, 2001), we could not find empirical evidence in the literature quantifying the severity of the problem. Therefore, we evaluated two PCFGs (Bikel, 2004; Charniak and Johnson, 2005) on edit detection and compared their performance to a state-of-the-art TAG-based edit detection system (Johnson et al., 2004). For this experiment, we evaluated edit detection accuracy on a per-word basis, where any tree terminal is considered an edit word if and only if it is dominated by an EDITED constituent in the gold tree. The PCFGs were trained on the train section of the treebank data with the flattened edit regions included[4] and then used to parse the test data.[5] The TAG-based detector was trained on the same conversation sides, with its channel model trained on the Penn Treebank disfluency-annotated files and its language model trained on trees with the EDITED nodes excised. As shown in Table 2, we did find that both PCFGs performed significantly below the TAG-based detector.

5 Results

In evaluating parse accuracy, we adopt the *relaxed edited* revision (Charniak and Johnson, 2001) to the standard PARSEVAL metric, which penalizes systems that get EDITED spans wrong, but does not penalize disagreements in the attachment or internal structure of edit regions. This metric is based on the assumption that there is little reason to recover syntactic structure in regions of speech that have been repaired or restarted by the speaker.

Table 3 shows the F-scores for the top-ranked parses after reranking, where the first-stage PCFG parser was run with a beam-size of 50. The first and last rows show lower and upper bounds, respectively, for reranked parsing accuracy on each edit condition. As the oracle rate[6] shows, there is con-

[4] Training on flattened EDITED nodes improved detection accuracy for both PCFGs: as much as 15% for Bikel-Collins.

[5] For the Charniak parser, edits were detected using only its PCFG component in 1-best mode, not its 2nd stage reranker.

[6] Oracle F uses the best parse in the 50-best list.

siderable room for improvement. Statistical significance was computed using a non-parametric shuffle test similar to that in (Bikel, 2004). For TAG and oracle *edit detection* conditions, the improvement from using the combined features over either prosodic or syntactic features in isolation was significant ($p < 0.005$). (For PCFG edit detection, $p < 0.04$.) Similarly, for all three *feature extraction* conditions, the improvement from using the TAG-based edit detector instead of the PCFG edit detector was also significant ($p < 0.001$). Interestingly, the TAG's 34% reduction in edit detection error relative to the PCFG yielded only about 23% of the parse accuracy differential between the PCFG and oracle conditions. Nevertheless, there remains a promising 2.0% difference in parse F-score between the TAG and oracle detection conditions to be realized by further improvements in edit detection. Training for the *syntactic* feature condition resulted in a learned weight λ with approximately 50K features, while the *prosodic* features used only about 1300 features. Despite this difference in the length of the λ vectors, the prosodic feature condition achieved 40–50% of the improvement of the syntactic features.

In some situations, e.g. for language modeling, improving the ordering and weights of the entire parse set (an not just the top ranked parse) is important. To illustrate the overall improvement of the reranked order, in Table 4 we report the reranked-oracle rate over the top s parses, varying the beam s. The error for each feature condition, relative to using the PCFG parser in isolation, is shown in Figure 3. Both the table and figure show that the reranked beam achieves a consistent trend in parse accuracy improvement relative to the PCFG beam, similar to what is demonstrated by the 1-best scores (Table 3).

Table 4: Reranked-oracle rate parse F-score for the top s parses with reference edit detection.

s	1	2	3	5	10	25
PCFG	86.9	89.8	91.0	92.2	93.4	94.6
Pros.	87.6	90.3	91.5	92.7	93.9	94.8
Syn.	88.4	91.3	92.4	93.4	94.3	95.0
Comb.	88.6	91.5	92.5	93.5	94.4	95.0

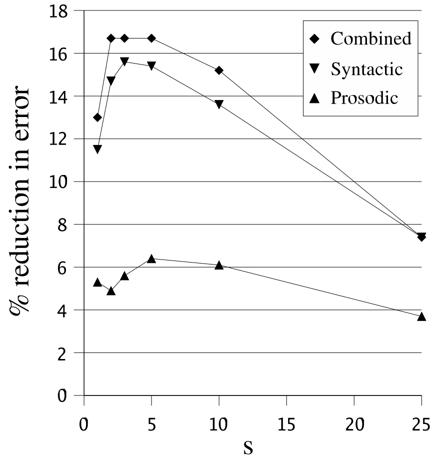

Figure 3: Reduction in error (Error $= 1 - F$) for the s-best reranked-oracle relative to the parser-only oracle, for different feature rerankings (reference edit detection).

6 Conclusion

This study shows that incorporating prosodic information into the parse selection process, along with non-local syntactic information, leads to improved parsing accuracy on accurate transcripts of conversational speech. Gains are shown to be robust to difficulties introduced by automatic edit detection and, in addition to improving the one-best performance, the overall ordering of the parse candidates is improved. While the gains from combining prosodic and syntactic features are not additive, since the prosodic features incorporates some constituent-structure information, the combined gains are significant. These results are consistent with related experiments with a different type of prosodically cued event, which showed that automatically detected IPs based on prosodic cues (Liu *et al.*, 2004) are useful in the reranking stage of a TAG-based speech repair detection system (Johnson et al., 2004).

The experiments described here used automatically extracted prosodic features in combination with human-produced transcripts. It is an open question as to whether the conclusions will hold for errorful ASR transcripts and automatically detected SU boundaries. However, there is reason to believe that relative gains from using prosody may be larger than those observed here for reference transcripts (though overall performance will degrade), based on prior work combining prosody and lexical cues to detect other language structures (Shriberg and Stolcke, 2004). While the prosody feature extraction depends on timing of the hypothesized word sequence, the acoustic cues are relatively robust to word errors and the break model can be retrained on recognizer output to automatically learn to discount the lexical evidence. Furthermore, if parse reranking operates on the top N ASR hypotheses, the reranking procedure can improve recognition outputs, as demonstrated in (Kahn, 2005) for syntactic features alone. Allowing for alternative SU hypotheses in reranking may also lead to improved SU segmentation.

In addition to assessing the impact of prosody in a fully automatic system, other avenues for future work include improving feature extraction. One could combine IP and prosodic break features (so far explored separately), find new combinations of prosody and syntactic structure, and/or incorporate other prosodic events. Finally, it may also be useful to integrate the prosodic events directly into the PCFG, in addition to their use in reranking.

This work was supported by the NSF under grants DMS-0074276, IIS-0085940, IIS-0112432, IIS-0326276, and LIS-9721276. Conclusions are those of the authors and do not necessarily reflect the views of the NSF.

References

A. Batliner *et al.* 1996. Prosody, empty categories and parsing - a success story. *Proc. ICSLP*, pp. 1169-1172.

J. Bear and P. Price. 1990. Prosody, syntax and parsing. *Proc. ACL*, pp. 17-22.

D. Bikel. 2004. *On the Parameter Space of Lexicalized Statistical Parsing Models*. Ph.D. thesis, U. Penn.

E. Charniak and M. Johnson. 2001. Edit detection and parsing for transcribed speech. *NAACL*, pp. 118-126.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. *Proc. ACL*.

M. Collins. 2000. Discriminative reranking for natural language parsing. *Proc. ICML*, pp. 175-182.

M. Core and L. Schubert. 1999. A syntactic framework for speech repairs and other disruptions. *Proc. ACL*, pp. 413-420.

D. Engel, E. Charniak, and M. Johnson. 2002. Parsing and disfluency placement. *Proc. EMNLP*, pp. 49-54.

D. Graff and S. Bird. 2000. Many uses, many annotations for large speech corpora: Switchboard and TDT as case studies. *Proc. LREC*, pp. 427-433.

M. Gregory, M. Johnson, and E. Charniak. 2004. Sentence-internal prosody does not help parsing the way punctuation does. *Proc. NAACL*, pp. 81-88.

P. A. Heeman and J. F. Allen. 1999. Speech repairs, intonational phrases, and discourse markers: Modeling speaker's utterances in spoken dialogue. *Computational Linguistics*, 25(4):527-571.

D. Hindle. 1983. Deterministic parsing of syntactic non-fluencies. *Proc. ACL*, pp. 123-128.

M. Johnson, E. Charniak, and M. Lease. 2004. An improved model for recognizing disfluencies in conversational speech. *Proc. Rich Transcription Workshop*.

J. G. Kahn, M. Ostendorf, and C. Chelba. 2004. Parsing conversational speech using enhanced segmentation. *Proc. HLT-NAACL 2004*, pp. 125-128.

J. G. Kahn. 2005. *Moving beyond the lexical layer in parsing conversational speech.* M.A. thesis, U. Wash.

LDC. 2004. Simple metadata annotation specification. Tech. report, Linguistic Data Consortium. Available at http://www.ldc.upenn.edu/Projects/MDE.

Y. Liu *et al.* 2004. The ICSI-SRI-UW metadata extraction system. *Proc. ICSLP*, pp. 577-580.

M. Meteer, A. Taylor, R. MacIntyre, and R. Iyer. 1995. Dysfluency annotation stylebook for the switchboard corpus. Tech. report, Linguistic Data Consortium.

E. Nöth *et al.* 2000. Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Trans. SAP*, 8(5):519-532.

M. Ostendorf *et al.* 2001. A prosodically labeled database of spontaneous speech. *ISCA Workshop on Prosody in Speech Recognition and Understanding*, pp. 119-121, 10.

J. Pitrelli, M. Beckman, and J. Hirschberg. 1994. Evaluation of prosodic transcription labeling reliability in the ToBI framework. *Proc. ICSLP*, pp. 123-126.

P. J. Price *et al.* 1991. The use of prosody in syntactic disambiguation. *JASA*, 90(6):2956-2970, 12.

E. Shriberg. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, U.C. Berkeley.

E. Shriberg and A. Stolcke. 2004. Prosody modeling for automatic speech recognition and understanding. *Mathematical Foundations of Speech and Language Processing*. Springer-Verlag, pp. 105-114.