Sentence-Internal Prosody Does not Help Parsing the Way Punctuation Does

Michelle L Gregory
Brown University
mgregory@cog.brown.edu

Mark Johnson
Brown University
Mark_Johnson@Brown.edu

Eugene Charniak
Brown University
ec@cs.brown.edu

Abstract

This paper investigates the usefulness of sentence-internal prosodic cues in syntactic parsing of transcribed speech. Intuitively, prosodic cues would seem to provide much the same information in speech as punctuation does in text, so we tried to incorporate them into our parser in much the same way as punctuation is. We compared the accuracy of a statistical parser on the LDC Switchboard treebank corpus of transcribed sentence-segmented speech using various combinations of punctuation and sentence-internal prosodic information (duration, pausing, and f0 cues). With no prosodic or punctuation information the parser's accuracy (as measured by F-score) is 86.9%, and adding punctuation increases its F-score to 88.2%. However, all of the ways we have tried of adding prosodic information decrease the parser's F-score to between 84.8% to 86.8%, depending on exactly which prosodic information is added. This suggests that for sentence-internal prosodic information to improve speech transcript parsing, either different prosodic cues will have to used or they will have be exploited in the parser in a way different to that used currently.

1 Introduction

Acoustic cues, generally duration, pausing, and f0, have been demonstrated to be useful for automatic segmentation of natural speech (Baron et al., 2002; Hirschberg and Nakatani, 1998; Neiman et al., 1998). In fact, it is generally accepted that prosodic information is a reliable tool in predicting topic shifts and sentence boundaries (Shriberg et al., 2000). Sentences are generally demarcated by a major fall (or rise) in f0, lengthening of the final syllable, and following pauses. However, the usefulness of prosodic information in sentence-internal parsing is less clear. While assumed not to be a one-to-one mapping, there is evidence that there is a strong correlation between prosodic boundaries and sentence-internal syntactic boundaries (Altenberg, 1987; Croft, 1995). For example, Schepman and Rodway (2000) have shown that prosodic cues reliably predict ambiguous attachment of relative clauses within coordination constructions. Jansen et al. (2001) have demonstrated that prosodic breaks and an increase in pitch range can distinguish direct quotes from indirect quotes in a corpus of natural speech.

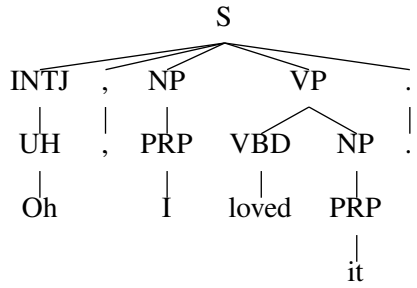

Figure 1: A treebank style tree in which punctuation is coded with terminal and preterminal nodes.

This paper evaluates the accuracy of a statistical parser whose input includes prosodic cues. The purpose of this study to determine if prosodic cues improve parsing accuracy in the same way that punctuation does. Punctuation is represented in the various Penn treebank corpora as independent word-like tokens, with corresponding terminal and preterminal nodes, as shown in Figure 1 (Bies et al., 1995). Even though this seems linguistically highly unnatural (e.g., punctuation might indicate suprasegmental prosodic properties), statistical parsers generally perform significantly better when their training and test data contains punctuation represented in this way than if the punctuation is stripped out of the training and test data (Charniak, 2000; Engel et al., 2002; Johnson, 1998). On the Switchboard treebank data set using the experimental setup described below we obtained an F-score of 0.882 when using punctuation and 0.869 when punctuation was stripped out, replicating previous experiments demonstrating the importance of punctuation. (F-score is a standard measure of parse accuracy, see e.g., Manning and Schütze (1999) for details).

This paper investigates how prosodic cues, when encoded in the parser's input in a manner similar to the way the Penn treebanks encode punctuation, affect parser accuracy. Our starting point is the observation that the Penn treebank annotation of punctuation does significantly improve parsing accuracy. Coupled with the assumption that punctuation and prosody are encoding similar information, this led us to try to encode prosodic information in a manner that was as similar as possible to the way that punctuation is encoded in the Penn treebanks.

For example, commas in text and pauses in speech seem to convey similar information. In fact, when transcribing speech, commas are often used to denote a pause. Thus, given the correlation between the two, and the fact that sentence-internal punctuation tends to be commas, we expected that pause duration, coded in a way similar to punctuation, would improve parsing accuracy in the same way that punctuation does.

While it may be the case that the encoding of prosodic information used in the experiments below is perhaps not optimal and the parser has not been tuned to use this information, note that exactly the same objections could be made to the way that punctuation is encoded and used in modern statistical parsers, and punctuation does in fact dramatically improve parsing accuracy.

We focus in this paper on parsing accuracy in a modern statistical parsing framework, but it is important to remember that prosodic cues might help parsing in other ways as well, even if they do not improve parsing accuracy. Nöth et al. (2000) point out that prosodic cues reduce parsing time and increase recognition accuracy when parsing speech lattices with the hand-crafted Verbmobil grammar. Page 266 of Kompe (1997) discusses the effect that incorporating prosodic information has on parse quality in the Verbmobil system using the TUG unification grammar parser: out of the 54 parses affected by the addition of prosodic information, 33 were judged "better with prosody", 14 were judged "better without prosody" and 7 were judged "unclear". Our experiments below differ from the experiments of Nöth and Kompe in many ways. First, we used speech transcripts rather than speech recognizer lattices. Second, we used a general-purpose broad-coverage statistical parser rather than a unification grammar parser with a hand-constructed grammar.

2 Method

The data used for this study is the transcribed version of the Switchboard Corpus as released by the Linguistic Data Consortium. The Switchboard Corpus is a corpus of telephone conversations between adult speakers of varying dialects. The corpus was split into training and test data as described in Charniak and Johnson (2001). The training data consisted of all files in sections 2 and 3 of the Switchboard treebank. The testing corpus consists of files sw4004.mrg to sw4153.mrg, while files sw4519.mrg to sw4936.mrg were used as development corpus.

2.1 Prosodic variables

Prosodic information for the corpus was obtained from forced alignments provided by Hamaker et al. (2003) and Ferrer et al. (2002). Hamaker et al. (2003) provided word alignments between the LDC parsed corpus and new alignments of the Switchboard Coprus. Most of the differences between the two alignments were individual lexical

items. In cases of differences, we kept the lexical item from the LDC version. Ferrer et al. (2002) provided very rich prosodic information including duration, pausing, f0 information, and individual speaker statistics for each word in the corpus. The information obtained from this corpus was aligned to the LDC corpus.

It is not known exactly which prosodic variables convey the information about syntactic boundaries that is most useful to a modern syntactic parser, so we investigated many different combinations of these variables. We looked for changes in pitch and duration that we expected would correspond to syntactic boundaries. While we tested many combinations of variables, they were mainly based on the variables PAU_DUR_N, NORM_LAST_RHYME_DUR, FOK_WRD_DIFF_MNMN_N, FOK_LR_MEAN_KBASELN and SLOPE_MEAN_DIFF_N in the data provided by Ferrer et al. (2002).

While Ferrer (2002) should be consulted for full details, PAU_DUR_N is pause duration normalized by the speaker's mean sentence-internal pause duration, NORM_LAST_RHYME_DUR is the duration of the phone minus the mean phone duration normalized by the standard deviation of the phone duration for each phone in the rhyme, FOK_WRD_DIFF_MNMN_NG is the log of the mean f0 of the current word, divided by the log mean f0 of the following word, normalized by the speakers mean range, FOK_LR_MEAN_KBASELN is the log of the mean f0 of the word normalized by speaker's baseline, and SLOPE_MEAN_DIFF_N is the difference in the f0 slope normalized by the speaker's mean f0 slope.

These variables all range over continuous values. Modern statistical parsing technology has been developed assuming that all of the input variables are categorical, and currently our parser can only use categorical inputs. Given the complexity of the dynamic programming algorithms used by the parser, it would be a major research undertaking to develop a statistical parser of the same quality as the one used here that is capable of using both categorical and continuous variables as input.

In the experiments below we binned the continuous prosodic variables to produce the actual categorical values used in our experiments. Binning involves a trade-off, as fewer bins involve a loss of information, whereas a large number of bins splits the data so finely that the statistical models used in the parser fail to generalize. We binned by first constructing a histogram of each feature's values, and divided these values into bins in such a way that each bin contained the same number of samples. In runs in which a single feature is the sole prosodic feature we divided that feature's values into 10 bins, while runs in which two or more prosodic features were conjoined we divided each feature into 5 bins.

While not reported here, we experimented with a wide variety of different binning strategies, including using the bins proposed by Ferrer et al. (2002). In fact the number of bins used does not affect the results markedly; we obtained virtually the same results with only two bins.

We generated and inserted "pseudo-punctuation" symbols based on these binned values that were inserted into the parse input as described below. In general, a pseudo-punctuation symbol is the conjunction of the binned values of all of the prosodic features used in a particular run. When mapping from binned prosodic variables to pseudo-punctuation symbols, some of the binned values can be represented by the absence of a pseudo-punctuation symbol.

Because we intend these pseudo-punctuation symbols to be as similar as possible to normal punctuation, we generated pseudo-punctuation symbols only when the corresponding prosodic variable falls outside of its typical values. The ranges are given below, and were chosen so that they align with bin boundaries and result in each type of pseudo-punctuation symbol occuring on 40% of words. Thus when a prosodic feature is used alone only 4 of its 10 bins are represented by a pseudo-punctuation symbol.

However, when two or more types of the prosodic pseudo-punctuation symbols are used at once there is a larger number of different pseudo-punctuation symbols and a greater number of words appearing with a following pseudo-punctuation symbol. For example, when P, R and S prosodic annotations are used together there are 89 distinct types of prosodic pseudo-punctuation symbols in our corpus, and 54% of words are followed by a prosodic pseudo-punctuation symbol.

The experiments below make use of the following types of pseudo-punctuation symbols, either alone

or concatenated in combination. See Figure 2 for an example tree with pseudo-punctuation symbols inserted.

P*b* This is based on the bin b of the binned PAU_DUR_N value, and is only generated when the PAU_DUR_N value is greater than 0.285.

R*b* This is based on the bin b of the binned NORM_LAST_RHYME_DUR value, and is only generated that value is greater than -0.061.

W*b* This is based on the bin b of the binned FOK_WRD_DIFF_MNMN_N value, and is only generated when that value is less than -0.071 or greater than 0.0814.

L*b* This is based on the bin b of the FOK_LR_MEAN_KBASELN value, and is only generated when that value is less than 0.157 or greater than 0.391.

S*b* This is based on the bin b of the SLOPE_MEAN_DIFF_N value, and is only generated whenever that value is non-zero.

In addition, we also created a binary version of the P feature in order to evaluate the effect of binarization.

NP This is based on the PAU_DUR_N value, and is only generated when that value is greater than 0.285.

We actually experimented with a much wider range of binned variables, but they all produced results similar to those described below.

2.2 Parse corpus construction

We tried to incorporate the binned prosodic information described in the previous subsection in a manner that corresponds as closely as possible to the way that punctuation is represented in this corpus, because previous experiments have shown that punctuation improves parser performance (Charniak and Johnson, 2001; Engel et al., 2002). We deleted disfluency tags and EDITED subtrees from our training and test corpora.

We investigated several combinations of prosodic pseudo-punctuation symbols. For each of these we generated a training and test corpus. The pseudo-punctuation symbols are dominated by a new preterminal PROSODY to produce a well-formed tree. These prosodic local trees are introduced into the tree following the word they described, and are attached as high as possible in the tree, just as punctuation is in the Penn treebank. Figure 2 depicts a typical tree that contains P R S prosodic pseudo-punctuation symbols inserted following the word they describe.

We experimented with several other ways of incorporating prosody into parse trees, none of which greatly affected the results. For example, we also experimented with a "raised" representation in which the prosodic pseudo-punctuation symbol also serves as the preterminal label. The corresponding "raised" version of the example tree is depicted in Figure 3.

The motivation for raising is as follows. The statistical parser used for this research generates the siblings of a head in a sequential fashion, first predicting the category label of a sibling and later conditioning on that label to predict the remaining siblings. "Raising" should permit the generative model to condition not just on the presence of a prosodic pseudo-punctuation symbol but also on its actual identity. If some but not all of the prosodic pseudo-punctuation symbols were especially indicative of some aspect of phrase structure, then the "raising" structures should permit the parsing model to detect this and condition on just those symbols. Note that in the Penn treebank annotation scheme, different types of punctuation are given different preterminal categories, so punctuation is encoded in the treebank using a "raised" representation.

The resulting corpora contain both prosodic and punctuation information. We prepared our actual training and testing corpora by selectively removing subtrees from these corpora. By removing all punctuation subtrees we obtain corpora that contain prosodic information but no punctuation, by removing all prosodic information we obtain the original treebank data, and by removing both prosodic and punctuation subtrees we obtain corpora that contain neither type of information.

2.3 Evaluation

We trained and evaluated the parser on the various types of corpora described in the previous section.

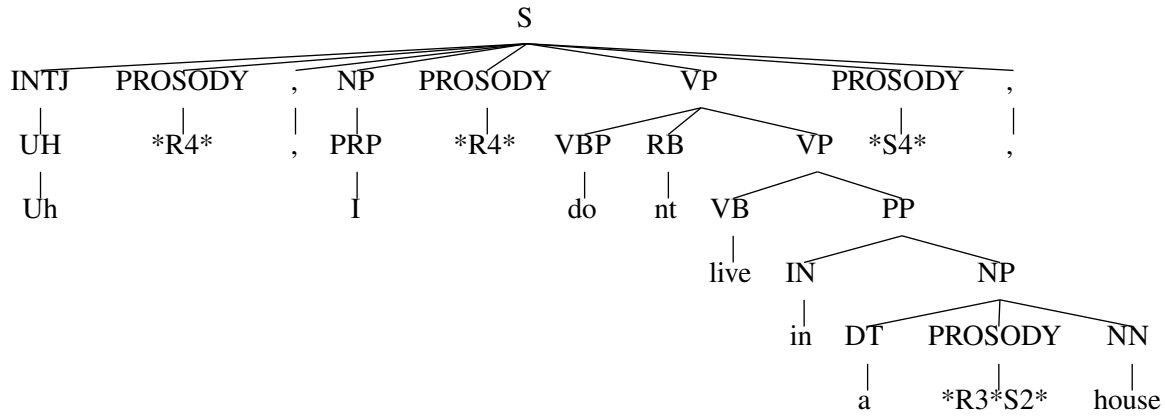

Figure 2: A tree with P R S prosodic pseudo-punctuation symbols inserted following the words they correspond to. (No P prosodic features occured in this utterance).

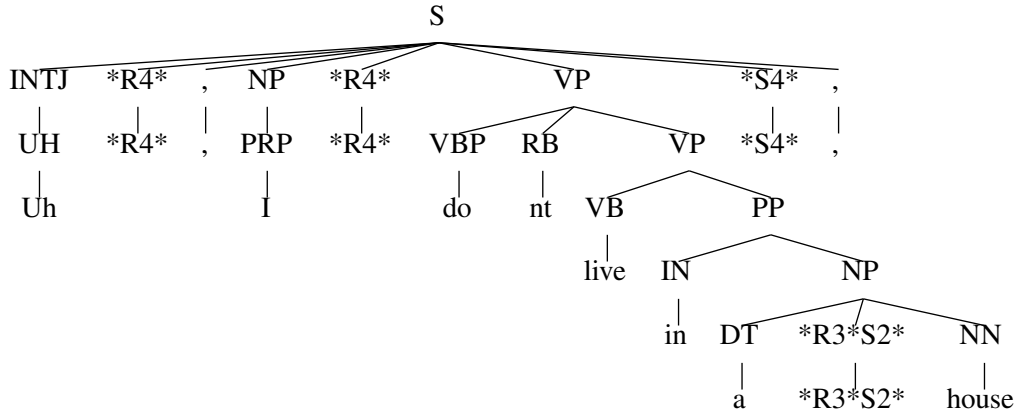

Figure 3: The same sentence as in Figure 2, but with prosodic pseudo-punctuation raised to the preterminal level.

Annotation	unraised	raised
punctuation	88.212	
none	86.891	
L	85.632	85.361
NP	86.633	86.633
P	86.754	86.594
R	86.407	86.288
S	86.424	85.75
W	86.031	85.681
P R	86.405	86.282
P W	86.175	85.713
P S	86.328	85.922
P R S	85.64	84.832

Table 1: The F-score of the parser's output when trained and tested on corpora with varying prosodic pseudo-punctuation symbols. The entry "punctuation" gives the parser's performance on input with standard punctuation, while "none" gives the parser's performance on input without any punctuation or prosodic pseudo-punctuation whatsoever.

(We always tested on the type of corpora that corresponded to the training data). We evaluated parser performance using the methodology described in Engel et al. (2002), which is a simple adaptation of the well-known PARSEVAL measures in which punctuation and prosody preterminals are ignored. This evaluation yields precision, recall and F-score values for each type of training and test corpora.

3 Results

Table 1 presents the results of our experiments. The RAISED prosody entry corresponds to the raised version of the COMBINED corpora, as described above.

We replicated previous results and showed that punctuation information does help parsing. However, none of the experiments with prosodic information resulted in improved parsing performance; indeed, adding prosodic information reduced performance by 2 percentage points in some cases. This is a very large amount by the standards of modern statistical parsers. Notice that the general trend is that performance decreases as the amount and complexity of the prosodic annotation increased.

4 Discussion and Conclusion

Simple statistical tests show that there is in fact a significant correlation between the location of opening and closing phrase boundaries and all of the prosodic pseudo-punctuation symbols described above, so there is no doubt that these do convey information about syntactic structure. However, adding the prosodic pseudo-punctuation symbols uniformly decreased parsing accuracy relative to input with no prosodic information. There are a number of reasons why this might be the case.

While we investigated a wide range of prosodic features, it is possible that different prosodic features might improve parsing performance, and it would be interesting to see if improved prosodic feature extraction would improve parsing accuracy.

We suspect that the decrease in accuracy is due to the fact that the addition of prosodic pseudo-punctuation symbols effectively excluded other sources of information from the parser's statistical models. For example, as mentioned earlier the parser uses a mixture of n-gram models to predict the sequence of categories on the right-hand side of syntactic rules, backing off ultimately to a distribution that includes just the head and the preceding sibling's category. Consider the effect of inserting a prosodic pseudo-punctuation symbol on such a model. The prosodic pseudo-punctuation symbol would replace the true preceding sibling's category in the model, thus possibly resulting in poorer overall performance (note however that the parser also includes a higher-order backoff distribution in which the next category is predicted using the preceding two sibling's categories, so the true sibling's category would still have some predictive value).

The basic point is that inserting additional information into the parse tree effectively splits the conditioning contexts, exacerbating the sparse data problems that are arguably the bane of all statistical parsers. Additional information only improves parsing accuracy if the information it conveys is sufficient to overcome the loss in accuracy incurred by the increase in data sparseness. It seems that punctuation carries sufficient information to overcome this loss, but that the prosodic categories we introduced do not.

It could be that our results reflect the fact that we

are parsing speech transcripts in which the words (and hence their parts of speech) are very reliably identified, whereas our prosodic features were automatically extracted directly from the speech signal and hence might be noisier. If the explanation proposed above is correct, it is perhaps not surprising that an accurate part of speech label would prove more useful in a conditioning context used by the parser than a noisy prosodic feature. Note that this would not be the case when parsing from speech recognizer output (since word identity would itself be uncertain), and it is possible that in such applications prosodic information would be more useful.

Of course, there are many other ways prosodic information might be exploited in a parser, and one of those may yield improved parser performance. We chose to incorporate prosodic information into our parser in a way that was similar to the way that punctuation is annotated in the Penn treebanks because we assumed that punctuation carries information similar to prosody, and it had already been demonstrated that punctuation annotated in the Penn treebank fashion does systematically improve parsing accuracy.

But the assumption that prosody conveys information about syntactic structure in the same way that punctuation does could be false. It could also be that even though prosody encodes information about syntactic structure, this information is encoded in a manner that is too complicated for our parser to utilize. For example, even though commas are often used to indicate pauses, pauses have many other functions in fluent speech. Pauses of greater than 200 ms are often associated with planning problems, which might be correlated with syntactic structure in ways too complex for the parser to exploit. While not reported here, we tried various techniques to isolate different functions of pauses, such as excluding pauses of greater than 200 ms. However, all of these experiments produced results similar to those reported here.

Finally, there is another possible reason why our assumption that prosody and punctuation are similar in their information content could be wrong. Our prosodic information was automatically extracted from the speech stream, while punctuation was produced by human annotators who presumably comprehended the utterances being annotated. Given this, it is perhaps no surprise that our automatically extracted prosodic annotations proved less useful than human-produced punctuation.

References

Bengt Altenberg. 1987. *Prosodic patterns in spoken English: studies in the correlation between prosody and grammar*. Lund University Press, Lund.

Don Baron, Elizabeth Shriberg, and Andreas Stolcke. 2002. Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 949–952, Denver.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre, 1995. *Bracketting Guideliness for Treebank II style Penn Treebank Project*. Linguistic Data Consortium.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 118–126.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *The Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

William Croft. 1995. Intonation units and grammatical structure. *Linguistics*, 33:839–882.

Donald Engel, Eugene Charniak, and Mark Johnson. 2002. Parsing and disfluency placement. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 49–54.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2002. Is the speaker done yet? faster and more accurate end-of-utterance detection using prosody in human-computer dialog. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 3, pages 2061–2064, Denver.

Luciana Ferrer. 2002. Prosodic features for the switchboard database. Technical report, SRI International, Menlo Park.

Jon Hamaker, Dan Harkins, and Joe Picone. 2003. Manually corrected switchboard word alignments.

Julia Hirschberg and Christine Nakatani. 1998. Acoustic indicators of topic segmentation. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 4, pages 1255–1258, Philadelphia.

Wouter Jansen, Michelle L. Gregory, and Jason M. Brenier. 2001. Prosodic correlates of directly reported speech: Evidence from conversational speech. In *Proceedings of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, pages 77–80, Red Banks, NJ.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Ralf Kompe. 1997. *Prosody in speech understanding systems*. Springer, Berlin.

Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Heinrich Neiman, Elmar Noth, Anton Batliner, Jan Buckow, Florian Gallwitz, Richard Huber, and Volkar Warnke. 1998. Using prosodic cues in spoken dialog systems. In *Proceedings of the International Workshop on Speech and Computer*, pages 17–28, St. Petersburg.

Elmar Nöth, Anton Batliner, Andreas Kießling, Ralf Kompe, and Heinrich Niemann. 2000. Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Transactions on Speech and Auditory Processing*, 8(5):519–532.

Astrid Schepman and Paul Rodway. 2000. Prosody and on-line parsing in coordination structures. *The Quarterly Journal of Experimental Psychology: A*, 53(2):377–396.

Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tur, and Gorkhan Tur. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154.