Exploiting Prosody for PCFGs with Latent Annotations

Markus Dreyer[1], *Izhak Shafran*[2]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2]Center for Spoken Language Understanding, OGI School of Science & Engg., Portland, OR, USA
markus@clsp.jhu.edu, zak@cslu.ogi.edu

Abstract

We propose novel methods for integrating prosody in syntax using generative models. By adopting a grammar whose constituents have latent annotations, the influence of prosody on syntax can be learned from data. In one method, prosody is utilized to seed the latent annotations of a grammar which is then refined using EM iterations. In an orthogonal approach, we integrate prosody into grammar more explicitly using a model that jointly observes words and associated prosody. We evaluate the two methods by parsing speech data from the Switchboard corpus. The results are compared against baseline results from a model that does not use prosody. The experiments show that prosody improves a grammar in terms of accuracy as well as the parsimonious use of parameters.

10.21437/Interspeech.2007-216

1. Introduction

The interaction between prosody and syntax is confounded by factors such as contrastive focus, introduction of new information, length of constituents and the availability of multiple options in conveying prosodic cues in production [1, 2]. Researchers, however, have been successful in uncovering certain characteristics of the interaction between prosody and syntax. For example, Pynte and colleagues studied the role of prosodic breaks in the parsing of locally ambiguous sentences such as "The spies inform the guards of the conspiracy" [3].

The recent availability of prosodically marked corpora of natural conversational speech such as [4] has spurred research on statistical approaches that use prosody to aid the parsing of spoken language (e.g. [5, 6]). This paper proposes a generative model to take advantage of prosodic cues in a probabilistic context free grammar (PCFG). The model skirts the issue of explicitly specifying the relationship between prosody and syntax by using latent annotations on the syntactic nonterminals. Following a brief review of related work in Section 2, we describe the model and its estimation in Section 3. Subsequently, in Section 4, we describe the experiments and report results of empirical evaluation on the Switchboard corpus. Finally, we analyze the results and discuss the impact of the model in Section 5.

2. Related Work

Several frameworks have been proposed to exploit prosodic information in parsing spoken utterances [7, 5, 6, 8]. Most early work was performed in the context of highly constrained tasks where it was used to rescore a set of candidate parses or to rule out certain hypotheses [7, 5]. Recent work utilized subsets of the Switchboard corpus which are manually annotated with syntactic and prosodic structure [8, 6].

Gregory et al. introduced various "pseudo-punctuation" symbols directly into the grammar, where the symbols encoded high values of acoustic measurements such as pause duration, rhyme duration, pitch, and pitch slope [8]. They observed a loss of performance, leading them to doubt the core assumption of their model that prosody conveys information in a manner similar to punctuation in text. An alternative approach utilized prosodic cues as features in a discriminative reranking framework [6]. This allowed the model to incorporate a wide variety of features including the association of prosody with the size of constituents, a property difficult to capture in a PCFG.

The present paper, unlike [6], proposes to incorporate prosody using a generative model. We avoid the pitfall, noted in [8], of specifying an explicit relationship between prosody and syntax; rather, the model infers the relationships from the data. Direct quantization of acoustic measurements severely limits the number of prosodic features that can be incorporated into the syntactic model. Instead, we use an intermediate symbol which is predicted from a rich set of prosodic cues.

3. Modeling Approach

The approach adopted in this paper for modeling prosody in syntax is comprised of three components. First, the level of disjuncture between a word and its subsequent neighbor is detected from the speech signal using a classifier that operates on a large number of prosodic features. Words are augmented with this automatically detected disjuncture. Second, the nonterminals are decorated with latent annotations to allow prosodic cues to propagate in the tree without explicitly defining their relationship. Third, the interface between syntax and prosody at the preterminal is modeled using two different methods.

3.1. Prosodic Breaks

The level of disjuncture is represented in terms of ToBI annotation, which assigns integer values from 0 to 4 with increasing severity of disjuncture [9]. In addition, a suffix p denotes perceptually disfluent events reflecting, for example, hesitation or planning. In conversational speech the intermediate levels occur infrequently and the break indices can be broadly categorized into three groups, namely, 1, 4 and p. A classifier was developed to predict three break indices at each word boundary based on per-speaker normalized variations in pitch, duration and energy associated with word, syllable or sub-syllabic constituents, as described in [10, 11]. A bagging-based break-index classifier was trained using a subset of Switchboard, manually annotated with ToBI labels. On ten-fold cross-validation, the classifier predicted prosodic breaks with an accuracy of 83.12% while chance was 67.66%. Each word is tagged with a symbol denoting the level of disjuncture following it. In this work, we chose to explore only two levels of disjuncture, corresponding to the automatically predicted break values of '1' or 'p'(b0) and

'4' (b1).

3.2. Latent Annotations

A straightforward approach for utilizing the prosodic tags is to model tagged sentences as samples from an enriched PCFG. A natural scheme would be to ensure that each PCFG nonterminal, just like each word, is tagged with the level of its following break. Modifying the training example `NP (NNP James`$_{b0}$`) (NNP Brown`$_{b1}$`)` in this way would yield the rewrite rule `NP`$_{b1}$ $\rightarrow$ `NNP`$_{b0}$ `NNP`$_{b1}$. The empirical probabilities may then reduce the likelihood of observing a prosodic break between the two words of a proper noun. Such a model was shown to improve detection of speech repairs in conversational speech [10].

One weakness of this model is that it encodes a simple explicit relationship between prosody and syntax, much like the deficiency noted in [8]. The true relationship is more complex and difficult to describe with hand-coded rules. Thus, instead of propagating tags deterministically, we will *learn* a scheme for enriching the nonterminals with tags. This allows the flexibility of percolating the prosodic breaks up the tree only in certain contexts or only to a certain level in the tree, all of which is learned from the data.

Several researchers have explored unsupervised learning techniques to modify the symbols in a grammar and tailor them for a given task. Matsuzaki et al. use PCFGs with latent annotations (PCFG-LA), splitting every nonterminal of a grammar G into L equally likely subsymbols, resulting in a grammar G_{LA} whose size is L^3 times the size of G [12]. If, for example, L is chosen to be 2, an original PCFG rule `X` $\rightarrow$ `Y Z` with probability .1 is, prior to training, substituted by eight rules.

$$
\begin{aligned}
X_0 &\rightarrow Y_0 Z_o \\
X_0 &\rightarrow Y_0 Z_1 \\
&\cdots \\
X_1 &\rightarrow Y_1 Z_0 \\
X_1 &\rightarrow Y_1 Z_1
\end{aligned}
$$

Each rule is initialized with uniform probability (.025) since `X`$_0$ and `X`$_1$ each have four possibilities to rewrite. Then, EM is applied to re-estimate the rule probabilities of this fine-grained grammar. The parse trees are observed, but the numbers that annotate the nonterminals are hidden. EM iteratively maximizes the likelihood of the complete, annotated training data. During the EM run, the rewrite probabilities move apart from their initial uniform state to learn differences between, for example, subject and object NPs. Parsing with such refined grammars obtains state-of-the-art results. For more thorough descriptions of PCFG-LA models, see [12], [13], [14] and [15].

We use Matsuzaki's PCFG-LA method as a baseline. To train the baseline models, we first apply a few preprocessing steps to the training data, which will be described shortly in Section 4.1, and read a small PCFG grammar G without annotations off the training data, using maximum likelihood estimation. Then we modify G by splitting every nonterminal into L equally likely subsymbols, as described above, and refine the estimates using EM, until the likelihood of the complete training data converges. In another variant, splits are performed gradually in stages of binary splits interspersed with several iterations of EM.

3.3. Integrating prosodic breaks

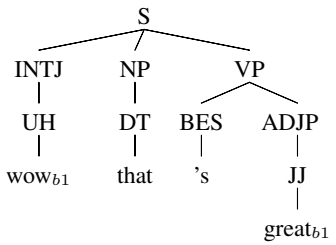
We propose two mechanisms for integrating prosodic breaks into a grammar with latent annotations. The first scheme employs modified observation probabilities for preterminals. The second one uses the prosodic breaks in the training input to initialize the annotated rewrite rules in a standard PCFG-LA without any additional distributions.

3.3.1. Modified observation probability

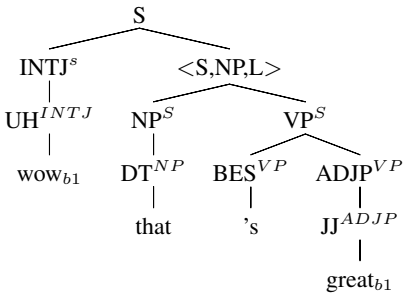
The first scheme (BRKOBS) extends the PCFG-LA model to explain jointly observed words and prosodic breaks. To alleviate data sparsity, we factorize the probability of a *lexical* rewrite rule `X`$_\alpha$ $\rightarrow$ `w`$_b$:

$$P(w_b \mid X_\alpha) = P(w \mid X_\alpha) P(b \mid X_\alpha)$$

where w is a word, b is its following prosodic break, X is a preterminal, and α is its latent annotation. This model assumes that the observation of a word, w, is conditionally independent of its prosodic break, b, given the preterminal with latent annotation, X_α. On a development set (see Section 4.1), this decomposition was found to perform better than other factorizations with factors such as $P(b \mid X)$, $P(b \mid \alpha)$, and $P(b \mid w, \alpha)$. The break probability, $P(b \mid X_\alpha)$, controls the degree to which a certain prosodic break b is associated with a certain annotated nonterminal.



(a) Training tree with prosodic breaks.



(b) The training tree, binarized and markovized.

Figure 1: All trees were modified by horizontal and vertical markovization and center-parent binarization. According to the head rules used [15], `VP` is the head of the rule `S` $\rightarrow$ `INTJ NP VP`. So, binarization subsumes the head `VP` and its neighbor `NP`. The complex auxiliary symbol `<S,NP,L>` encodes the original parent (`S`), the direction in which a child was subsumed moving outward from the head (`L` for left), and the most recent child subsumed (`NP`).

3.3.2. Initialization of latent annotations

An alternative approach (BRKINIT) is to utilize prosodic breaks to provide a more accurate initial grammar with latent annotations, G_{LA}. We percolate the break indices to each internal

node from its rightmost child, then read the grammar G_{LA} off these modified annotated training trees. This provides an initial model for the grammar associated with the non-terminals. After the initialization, the break indices are discarded from the fringe and the grammar with latent annotations is refined with EM. In a way, this model mimics "prosodic-bootstrapping", a hypothesis among psycholinguists that prosody plays an important initial role in syntax acquisition [16].

As an example, consider the parse tree in Figure 1b. An annotated lexical rewrite rule $\texttt{JJ}_1 \rightarrow \texttt{great}$ can be read off, using the break $b1$ as a latent annotation 1 on the nonterminal JJ. The parent annotation is here left out, for clarity. Similarly, $\texttt{ADVP}_1 \rightarrow \texttt{JJ}_1$ and $\texttt{VP}_1 \rightarrow \texttt{BES}_0\ \texttt{ADJP}_1$ are added to the grammar. The VP in the rule carries the latent annotation 1 since the last word in its yield, *great*, ends with a prosodic break $b1$. In every rewrite rule, the symbol on the left-hand side and the rightmost symbol on the right-hand side agree. This is effectively similar to the *pass to head* model constraint in [15] when the right-most child of every phrase is treated as its "head".

The described percolation of prosodic breaks up the tree is done to initialize G_{LA}. Once this initial grammar is constructed, its parameters are re-estimated as in standard PCFG-LA, using the parse trees as observed data and the annotations as hidden data. Note, this model tests how an otherwise unchanged PCFG-LA can be improved through prosody-driven initialization without introducing additional parameters. The decoding is performed on word sequences without prosodic tags.

The initialization procedure yields a PCFG-LA grammar G_{LA} with L=2. Once the EM re-estimation for the parameters of this grammar converges, the grammar is split again (L=2×2), without reference to breaks, and learning continues. In the first stage of training, with L=2, grammar rules have the constrained form $\texttt{X}_\alpha \rightarrow \texttt{Y}_\beta\ \texttt{Z}_\alpha$, as in $\texttt{VP}_1 \rightarrow \texttt{BES}_0\ \texttt{ADJP}_1$ described above. However, in the subsequent stages ($L = 2 \times 2$) the latent annotations on nonterminals may become decoupled from the breaks immediately following those nonterminals and unconstrained rules are possible. A split of $\texttt{ADJP}_1$ into $\texttt{ADJP}_1$ and $\texttt{ADJP}_2$, for example, makes an unconstrained rule like $\texttt{VP}_1 \rightarrow \texttt{BES}_0\ \texttt{ADJP}_2$ possible.

4. Experiments

4.1. Setup

We use transcribed speech data from the Switchboard treebanks as provided by the Linguistic Data Consortium. The partition into training, development, and test data is similar to the one in [17]. The training data was modified by four automated preprocessing procedures. We apply parent annotation, center-parent binarization[1] and vertical markovization [12], and replace low-frequency words (count < 3) with special markers. We apply additive lambda smoothing (λ=0.1) during training.

For decoding, we first create a 1000-best list of parses for each sentence, using a simple PCFG parser with no latent annotations. Each PCFG parse is rescored by the PCFG-LA, using a dynamic program to sum the probabilities of all ways of enriching its nonterminals with latent annotations [12]. The PCFG parse with the highest total probability is returned as the output.

4.2. Results

The results are shown in Figure 2. They illustrate the trade-offs between the number of parameters in the model and the parse accuracy. The baseline model, PCFG-LA, contains no prosodic information, and it is compared with two alternative methods BRKOBS and BRKINIT for incorporating prosodic information.

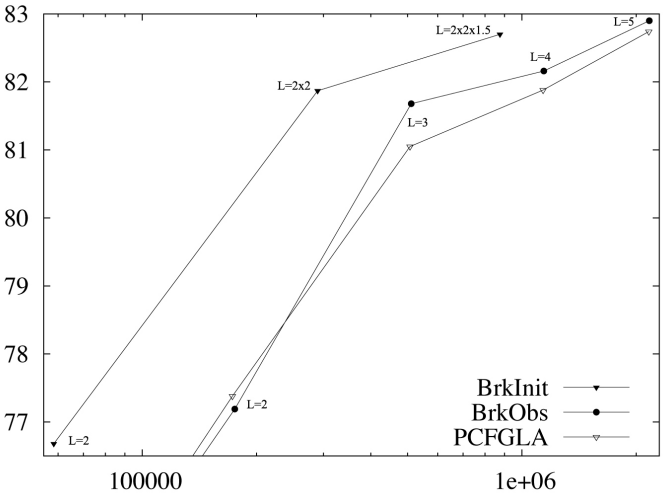


Figure 2: Numbers of parameters (x) vs F-score parsing results (y). BRKOBS and BRKINIT use prosodic breaks, while the PCFG baseline does not. Legend 2x2x1.5 corresponds to the case where half of the symbols in 2×2 are split again.

Modeling prosody explicitly in the observation (BRKOBS) improves performance over the baseline model (PCFG-LA) for all sizes of the model except L=2. This gain comes with only a small increase in parameters with respect to PCFG-LA. However, in this task, the improvements are statistically significant only at L=3, with p<.005. Anecdotally, the impact of such prosodic breaks can be seen on an example from the development set and its analyzes in Figure 3. Unlike PCFG-LA, BRKOBS exploits the prosodic breaks in the input and outputs the correct analysis.

Input: *i think i would have said that*$_{b1}$ *a few years ago*

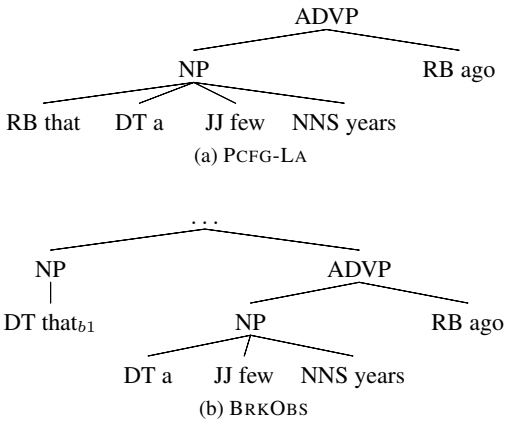


Figure 3: Anecdotal example from development set and the fragments of its analyzes by PCFG-LA and BRKOBS models.

A vanilla PCFG with no latent annotations reaches an F-score of only 71.67. A BRKOBS model with no latent anno-

[1] Unary rules are allowed.

tations, or L=1, reaches a similar performance, 71.76. The prosodic splits (BRKINIT) directly improve the score to 74.08 (before EM), then 76.68 (after EM, see L=2). It is known that prosodic and syntactic segmentations do not always align. However, these results suggest that prosodic breaks can provide a good initial guess for splitting syntactic nonterminals and help guide EM iterations to estimate a better latent-annotated syntactic model.

BRKINIT generally has the effect that parameters are greatly reduced, resulting in faster decoding times, with surprisingly good results. It obtains an F-score of 81.87 with under 290,000 parameters, while the baseline method PCFG-LA with default uniform initialization needs nearly four times as many parameters to reach similar parsing performance. BRKINIT obtains an F-score of 82.24 after splitting the annotations twice again.

Whether the nodes were split in one step or a sequence of staged splits, the grammars performed at about the same accuracy. The final test set showed the same trend as the development set with PCFG-LA performing at 82.77 and BRKOBS at 82.92 at L=5. BRKINIT obtains 82.72 on the final test set.

The presented prosody-aware methods obtain improvements in parsing accuracy and significant reductions in model size, as compared to a purely syntactic standard PCFG-LA baseline model. We believe that the parsing accuracies of all methods described in this paper can still be improved by using more nonterminal splits ($L > 5$), more sophisticated splitting methods and better smoothing [14, 17].

5. Summary

In this paper, we have investigated two mechanisms for exploiting prosodic breaks using grammars with latent annotations. By modifying the observation probabilities of preterminals to include prosodic tags (BRKOBS), we demonstrate consistent gains across latent annotations of size L=3, 4, 5. In an alternative integration of prosody, we show that a grammar split using prosodic criteria (BRKINIT) is a good first guess and leads the learner in the right direction. Starting the learner from such an initialization reduces the number of parameters needed to achieve equivalent performances.

6. Acknowledgments

We thank Jason Eisner for helpful discussions and suggestions.

7. References

[1] E. O. Selkirk, *Phonology and syntax: The relationship between sound and structure*. MIT Press, 1984.

[2] A. Cutler, D. Dahan, and W. van Donselaar, "Prosody in the comprehension of spoken language: A literature review," *Language and speech*, pp. 141–201, 1997.

[3] J. Pynte and B. Prieur, "Prosodic breaks and attachment decisions in sentence parsing," *Language and cognitive processes*, pp. 165–191, 1996.

[4] M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, L. Carmichael, and W. Byrne, "A prosodically labeled database of spontaneous speech," in *Proc. of ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, 2001, pp. 119–121.

[5] A. Batliner, A. Feldhaus, S. Geissler, T. Kiss, R. Kompe, and E. Nöth, "Prosody, empty categories and parsing – a success story," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1996, pp. 1169–1172.

[6] J. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf, "Effective use of prosody in parsing conversational speech," in *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005, pp. 233–240.

[7] N. M. Veilleux and M. Ostendorf, "Parse scoring with prosodic information," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1993, pp. 51–54.

[8] M. Gregory, M. Johnson, and E. Charniak, "Sentence-internal prosody does not help parsing the way punctuation does," in *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association of Computational Linguistics Annual Meeting (HLT/NAACL)*, 2004, pp. 81–88.

[9] H. F. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirshberg, "ToBI: A standard for labeling English prosody," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, vol. 2, 1992, pp. 867–870.

[10] J. Hale, I. Shafran, L. Yung, B. Dorr, M. Harper, A. Krasnyanskaya, M. Lease, Y. Liu, B. Roark, M. Snover, and R. Stewart, "PCFGs with syntactic and prosodic indicators of speech repairs," in *Proceedings of the joint conference of the International Conference on Computational Linguistics and the Association of Computational Linguistics (COLING/ACL)*, 2006, pp. 161–168.

[11] M. Harper, B. Dorr, J. Hale, B. Roark, I. Shafran, M. Lease, Y. Liu, M. Snover, and L. Yung, "Parsing and spoken structural event detection," in *2005 Johns Hopkins Summer Workshop Final Report*, 2005.

[12] T. Matsuzaki, Y. Miyao, and J. Tsujii, "Probabilistic CFG with latent annotations," in *Proceedings of the ACL Conference*, 2005.

[13] D. Prescher, "Head-driven PCFGs with latent-head statistics," in *Proceedings of the International Workshop on Parsing Technologies (IWPT)*, 2005, pp. 115–124.

[14] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proceedings of the ACL Conference*, 2006, pp. 433–440.

[15] M. Dreyer and J. Eisner, "Better informed training of latent syntactic features," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 317–326.

[16] L. A. Gerken, P. W. Jusczyk, and D. R. Mandel, "When prosody fails to cue syntactic structure: 9-month-olds' sensitivity to phonological versus syntactic phrases," *Cognition*, no. 51, pp. 237–265, 1994.

[17] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Proceedings of the North American Chapter of the Association of Computational Linguistics Annual Meeting (NAACL)*, 2001.