Where should pitch accents and phrase breaks go? A syntax tree transducer solution

Joseph Tepperman and Emily Nava

Rosetta Stone Labs
{jtepperman,enava}@rosettastone.com

Abstract

Motivated by a desire to assess the prosody of foreign language learners, this study demonstrates the benefit of high-level syntactic information in automatically deciding where phrase breaks and pitch accents should go in text. The connection between syntax and prosody is well-established, and naturally lends itself to tree-based probabilistic models. With automatically-derived parse trees paired to tree transducer models, we found that categorical prosody tags for unseen text can be determined with significantly higher accuracy than they can with a baseline method that uses n-gram models of part-of-speech tags. On the Boston University Radio News Corpus, the tree transducer outperformed the baseline by 14% overall for accents, and by 3% overall for breaks. These automatic results fell within this corpus's range of inter-speaker agreement in assigning accents and breaks to text.

Index Terms: syntax, prosody, ToBI, TTS, tree transducers

10.21437/Interspeech.2011-447

1. Introduction

Teaching categorical prosody to language learners - i.e. teaching them which words in a phrase should be accented or should mark a phrase break - can be a tricky problem for many reasons. Not the least of these reasons is the multiplicity of acceptable forms for accents and phrase breaks within a given piece of English text [2]. Take, for example, "Massachusetts got tough on drunken drivers years ago." This sentence was read aloud by five different professional newscasters in the Boston University Radio News Corpus (BURNC) [7]. Each of these speakers can be considered an expert in producing native English prosody. Yet, according to the Tones and Break Indices (ToBI) labeling convention with which the corpus has been annotated [9], each speaker realized the text with unique divisions into intermediate (|) and final (‖) intonational phrases, and also spoke the words with a unique distribution of **pitch accents**:

Massachusetts *got* ***tough*** | *on* ***drunken drivers*** | ***years*** *ago.* ‖
Massachusetts | *got* ***tough*** *on drunken drivers* ‖ ***years*** *ago.* ‖
Massachusetts *got* ***tough*** *on* ***drunken drivers*** | ***years ago***. ‖
Massachusetts *got* | ***tough*** *on* ***drunken*** *drivers* ***years*** *ago.* ‖
Massachusetts | ***got tough*** | *on* ***drunken drivers years ago***. ‖

If we assume that each word can precede a break and/or take on an accent, then this eight-word sentence theoretically has $4^8 = 65,536$ different allowable realizations. Given this massive amount of potential variability among native speakers, how can we hope to assess the prosody of a second-language student of English?

In cases such as these, a syntactic parse of the sentence can reveal underlying agreement in forms among these speakers:

```
S(NP(NNP:Massachusetts) VP(VP(VBD:got
ADJP(JJ:tough PP(IN:on NP(JJ:drunken
NNS:drivers)))) ADVP(NNS:years RB:ago)))
```

Here we see that in the surface forms produced by the five speakers, sentence-internal breaks tend to fall at the beginnings and ends of syntactic phrases, rather than within them. Also note that all words are not equal candidates for pitch accents: they are unlikely to be assigned to a preposition like "on," but should be expected on proper nouns ("Massachusetts") and adjectives ("tough"). With enough training examples, these preliminary observations can be generalized into probabilistic rules that can be used to predict breaks and accents in previously unseen text. Such a method can be used to create a reference standard of native prosody for teaching language learners.

This connection between syntax and prosody should not come as a surprise. The theory behind the ToBI system of prosodic transcription lends itself naturally to a hierarchical representation of intonational phrases nested within intonational phrases, corresponding roughly to the syntactic phrases delineated by parse trees [8]. ToBI is a powerful system that allows researchers to share a common vocabulary to describe the prosodic landscape of an utterance. Prominence is indicated with pitch accents that occur on stressed syllables, and phrasing labels indicate the flow pattern of a speaker's production. The power of this system lies in its ability to capture the nature of connected speech in ongoing discourse, which includes disfluencies and variable patterns of emphasis. And the link from prosodic theory to syntactic structure is not unique to ToBI. This approach mirrors traditional efforts in generative syntax to account for prominence assignment and distribution on the basis of embedding and structural hierarchy [3].

In the field of text-to-speech (TTS) synthesis, the use of syntactic information to assign natural phrase breaks to text has been exploited to some lengths. Dominant past approaches have either been rule-based, focusing primarily on constituency relationships such as verb adjacency [1], or data-driven approaches working with n-gram models of POS tags and their associated prosodic labels. Other data-driven studies have also incorporated constituency features into non-sequential models [4].

The present study is a continuation of the work in [11], which first introduced the idea of representing sequences of ToBI boundary tones and accents as trees, modeled with Regular Tree Grammars (RTGs) [6]. That study found that tree models could predict a missing ToBI label in a sequence of such labels better than an n-gram model could, a first indication of the explanatory power of the tree representation. However, syntactic information was conspicuously absent from that study, making it difficult to adapt it to tasks like break prediction from text. Furthermore, while the ToBI system does naturally lend itself to tree-like representations, the labels themselves were not

	train	*development*	*test*
speakers	3	2	2
sentences	771	128	89
words + punctuation	16,712	2721	1753

Table 1: Sizes of datasets used in this study.

originally derived to be treated like syntactic constituents - they are tones or breaks that occur at fixed points in time, not phrase-level units like the VPs and NPs of syntactic parse trees [8].

In this work, we intend to predict ToBI pitch accents and phrase breaks from text alone using Weighted Regular Tree Transducers (WRTTs). WRTTs are probabilistic models normally used in syntax-based machine translation [6]. In this case, they are trained to "translate" syntax trees (derived directly from text with an automatic parser) into one-level "trees" (i.e. sequences) of the most likely corresponding ToBI labels. This method will be compared with the standard n-gram POS model outlined in [10], in an effort to show the benefit of constituency information above the POS level, as well as the benefits of the tree transducer model. Such a system can function as one component in a complete TTS system, generating the labels from which the acoustics of synthetic speech are derived, or it can work as part of an automatic prosodic assessment routine - the accents and breaks determined here can serve as a reference against which to compare an ESL student's production. Our hypothesis is that a tree-based approach will maximally account for prosodic structure, where pitch accent and phrase break classification will benefit from knowledge of syntactic grouping.

2. Speech Data

The Boston University Radio News Corpus (BURNC) [7] consists of recordings from seven professional newscasters reading real radio news stories both in a laboratory setting and on the radio. Data from five of these speakers were annotated with ToBI labels for tones and breaks, and the transcripts also include part-of-speech tags, hand-corrected from an automatic tagger. For the present study, the data from the five prosodically-annotated speakers were divided into train, development, and test sets. The sizes of these sets are given in Table 1. The dev and test sets had unique speakers not seen in the train set, and the test set consisted only of news stories not seen in the train or dev sets (though the train and dev sets shared some news stories in common). To obtain syntactic trees of the text transcripts, we used the lexicalized version of the Stanford Parser [5] with its default settings (i.e. we did not retrain it). This parser is reported to have an F-score of 86.36% on the Penn Treebank corpus, and we found that it was able to predict 93.17% of the BURNC's hand-corrected POS tags (omitting punctuation, of course).

Some idea of the inter-speaker agreement in assigning breaks and accents to text was intimated in Section 1. The average pairwise speaker agreement was 78.72% (std = 4.75%) for accents and 84.27% (std = 3.43%) for breaks (if a break is defined as a ToBI index of 3 or above). This gives an upper bound on the level of performance we can expect for the automatic methods explored here. The true agreement may be higher or lower than these figures, considering that the inter-annotator agreement for ToBI labels on the BURNC is reported to be 91%, in terms of word-level presence vs. absence [7].

3. Predicting Prosodic Tags

3.1. Previous Work

The prior work that this paper builds on [11] used tree grammars to model hierarchical structures of prosodic tones. Formally, a Probabilistic Context Free Grammar (PCFG) specifies a set of terminal and nonterminal symbols for which, beginning with a starting symbol, a sequence of probabilistic tree production rules for replacing the nonterminal symbols can be performed [6]. The probability of a tree T is the product of the probabilities of the n production rules $\alpha \rightarrow \beta$ that generated it,

$$P(T) = \prod_{i=1}^{n} P(\alpha_i \rightarrow \beta_i | \alpha_i)$$

This of course assumes all rules (and all subtrees generated by those rules) are independent, allowing for the versatility of modeling a larger tree implicitly through a sequence of smaller tree production rules. A Weighted Regular Tree Grammar (WRTG) is a finite-state acceptor of PCFG trees, representing all nonterminal symbols through states in the recognition network. Probabilities of production rules (i.e. state transitions through the WRTG) are estimated from the training data as

$$\hat{P}(\alpha \rightarrow \beta | \alpha) = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

where $Count(\alpha \rightarrow \beta)$ and $Count(\alpha)$ are the occurrences of the production rule $\alpha \rightarrow \beta$ and the symbol α, respectively.

As it was originally proposed, this tree model would follow the nested structure of information encoded by the ToBI tones: pitch accents nested inside of intonational phrases, and intermediate phrase boundary tones nested within phrase-final tones (see [11] for an illustration). But the shortcomings of this past approach lie in its computational complexity and in its removal from syntax.

3.2. Proposed Model

A Weighted Regular Tree Transducer (WRTT) is a probabilistic set of rules for top-down transformation of a Regular Tree grammar (as defined in Section 3.1) into another form of Regular Tree Grammar [6]. Each transformation rule is modeled as independent of all other rules, and so the best transformation of an input tree is the one that maximizes the product of all the transformation rules that might be applied to that tree. By removing tree constituents in the transduction process, a tree transducer can also transform a tree into a string of terminal symbols. Such a set of transformation rules can then be used to turn syntactic parse trees into the corresponding most likely strings of pitch accents or break indices.

As an illustration, let's look at a piece of the running example sentence: "on drunken drivers." We can obtain this phrase's parse tree automatically from the text alone, using the Stanford Parser [5]. The true syntax tree looks like this: `PP(IN NP(JJ NNS))` - a prepositional phrase (`PP`) with a preposition (`IN`) and a noun phrase (`NP`) inside of it. This tree can then be fed into the transducer, which will convert it into a string of prosodic tags. In the conventions of the tree automata toolkit Tiburon [6], the first top-down transduction rule might be `q.PP(x0:IN x1:NP)` $\rightarrow$ `A q.x1` - this means, for a subtree headed by constituent `PP`, with children `IN` and `NP`, this can be transformed into the sequence `A q.NP` - a pitch accent (`A`) followed by a sub-tree headed by a noun phrase. In transforming an input syntax tree into a string of pitch accent labels,

the above production rule's probability would be compared with that of `q.PP(x0:IN x1:NP)` → `N q.x1`, in which the preposition (`IN`) takes no pitch accent (`N`). The `q.NP` subtree with children `JJ` (adjective) and `NNS` (plural noun) is then transformed with one of four possible rules: `q.NP(x0:JJ x1:NNS)` → `N A`, or `q.NP(x0:JJ x1:NNS)` → `A A`, etc. Each terminal POS symbol in the tree is transformed into one and only one prosodic tag. For deeper trees, this transformation process continues by applying rule after rule until no more nonterminal symbols remain. To constrain the number of rules in the model, we will use separate transducers for breaks and accents.

This method makes use of higher-order syntactic information (i.e. above the level of the part-of-speech tag) that could help determine a prosodic tag. By transforming sub-trees into strings, the model preserves much of the original tree structure. In comparison, the baseline method (reproduced below) uses only POS tags (no syntactic constituents) with sequential models. Our previous work (above) used tree grammars but no syntax. And the work in [4] (not reproduced here) used higher-level syntactic information outside of a tree model. The novelty of the proposed approach is its use of tree models, on the assumption that they are suited for tree-based syntactic features.

3.3. Baseline Method

The baseline method in [10] proposed assigning phrase breaks to text by using an n-gram model for sequences of junctures between words combined with a Markov model for a sequence of POS tags given a corresponding juncture. The most likely sequence of breaks could then be decoded from a sequence of POS tags using a composition of these two networks and the Viterbi algorithm, in a "noisy channel" framework.

Formally, if every adjacent pair of POS tags c_k and c_{k+1} has a juncture $j_k = \{B, N\}$ (break or non-break) between them, then the POS Markov model is defined as a sequence of juncture states that emit probabilities of POS sequences, $P(C_k|j_k)$, where by definition the POS context is constrained to $C_k = c_{k-1}, c_k, c_{k+1}$. Since this only models a limited window of POS tags, the n-gram model for the sequence of junctures is defined as the set of probabilities:

$$P(j_k|J_{k-1}^N) = P(j_k|j_{k-1}, j_{k-2}, \ldots, j_{k-N+1})$$

where N is the order of the model (experiments in [10] found $N = 6$ to perform best). The probability of a juncture j_k given the POS context C_k and the previous N junctures is then approximated using Bayes' Rule:

$$P(j_k|J_{k-1}^N, C_k) \propto P(j_k|J_{k-1}^N) \cdot P(C_k|j_k)$$

This baseline approach was previously only applied to assigning phrase breaks to text [10]. For comparison with the present experiments with pitch accents, the baseline method was extended as follows: we simply substitute the word variable $w_k = \{A, N\}$ (accent or no accent) for j_k in the equations above. Note that, unlike the novel method proposed above, this baseline doesn't include information about phrase boundaries, so the grouping of "on drunken drivers" into a prepositional phrase (which might help determine a phrase break) is lost.

4. Experiments

The point of the experiments in this section is to show the performance of the new method from Section 3.2 in comparison to a sequential baseline from Section 3.3 - it is a comparison of high-level syntactic features in a tree model, versus low-level syntactic features in a sequential model. There are a number of elements in common between the two methods, in order to make this a fair comparison. All experiments used the same training, development, and test text, as well as the same part-of-speech tag set from the Staford Parser. Punctuation symbols were included in the training sets and models, but were omitted from calculations in the final results. Following the conventions of [10], breaks were defined as any transcribed ToBI break index of 3 or above, and all ToBI pitch accents (whether high or low or otherwise) were combined into one "accent" category. Experiments on pitch accents and phrase breaks were conducted and evaluated separately. POS tags for all data came from the Stanford Parser and the text alone. Baseline models were implemented as finite-state networks using Carmel, and all tree transducers were implemented in Tiburon [6]. All experiments were evaluated using the metrics defined in [10] - % breaks (or accents) correct, % non-breaks (or non-accents) correct, and % overall junctures (or words) correct.

	model	*accents*	*non-accents*	*overall*
tree transducer	*PI*	**82.74**	88.63	80.20
	PD	82.61	89.09	80.58
	GD	81.97	**90.12**	**81.28**
n-gram order	*1*	**74.29**	79.26	66.67
	2	73.55	**81.42**	**68.47**
	3	73.14	81.34	68.19
	4	72.65	80.30	66.91
	5	74.04	80.70	67.97
	6	74.12	80.66	67.99
	7	72.65	80.14	66.75
	8	73.14	79.34	66.19

Table 2: Pitch accent % accuracy on the dev set.

	model	*breaks*	*non-breaks*	*overall*
tree transducer	*PI*	**68.33**	89.64	81.69
	PD	67.94	90.45	82.37
	GD	67.71	**91.58**	**83.40**
n-gram order	*1*	**75.24**	83.81	77.58
	2	70.00	**85.61**	**78.06**
	3	71.75	85.17	**78.06**
	4	72.38	84.49	77.54
	5	71.75	84.57	77.46
	6	72.70	84.65	77.78
	7	73.49	84.33	77.66
	8	72.38	84.45	77.50

Table 3: Phrase break % accuracy on the dev set.

The baseline n-gram models for sequences of breaks/non-breaks (or accents/non-accents) were trained using the SRILM toolkit with the default Good-Turing smoothing and backoff. On the baseline POS Markov model that converted POS contexts to ToBI labels, we used the same backoff method described in [10] - for the probability of any sequence $P(c_{k-1}, c_k, c_{k+1}|j_k)$ not seen in the train set, we backed off to $P(c_k, c_{k+1}|j_k)$ (or, if this was unavailable, to $P(c_k|j_k)$). Similarly, backoff on the tree transducers involved including rules such that, if a previously unseen tree were encountered, the transducer model would still transform it, but without maintaining the full original sub-tree structure. For example, if the training set derived the rule `q.PP(x0:IN x1:NP)` → `N q.x1`, its backoff rule would look like `q.PP(x0: x1:)` → `q.x0 q.x1` - i.e. a sub-tree state headed by a prepositional phrase with unknown children is transformed into states headed by those children, whatever they are. This necessitated the creation of many tree-independent rules for terminal symbols, e.g. `q.IN` → `A`. After training, the probabilities of these

model	*accents*	*non-accents*	*overall*
GD tree transducer	83.45	85.43	76.83
2nd-order n-gram	65.60	80.60	62.72

Table 4: Pitch accent % accuracy on the test set.

model	*breaks*	*non-breaks*	*overall*
GD tree transducer	60.10	91.47	81.43
2nd-order n-gram	74.78	85.13	78.64

Table 5: Phrase break % accuracy on the test set.

backoff rules were found to be too high - after all, they represent more general cases of the specific tree rules. And so the trained backoff probabilities were divided by some factor to maximize the complete model's performance on the dev set (a factor of 10,000 was empirically found to be adequate).

Certain parameters of the models were tuned on the development set, which consisted entirely of sentences already seen in the training set, but uttered by new speakers. To find the best baseline model, the order of the baseline n-gram models for break or accent sequences was varied from 1 to 8 (as done in [10]) on the dev set. Similarly, three different tree models were evaluated on the dev set: parent-independent models (PI), parent-dependent models (PD), and grandparent-dependent models (GD). PI models use the ordinary transduction rules as outlined in Section 3.2 (e.g. `q.PP(x0:IN x1:NP)` $\rightarrow$ `A q.x1`), in which each non-terminal constituent in the tree is independent of its parent constituent one level above it in the tree. PD models change these rules to incorporate information about the parent constituents of all non-terminal constituents (e.g. `q.PP|ADJP(x0:IN x1:NP|PP)` $\rightarrow$ `A q.x1` - this incorporates deeper tree information in the syntactic features for decoding. The GD trees take it one step further by incorporating parents and grandparents of each constituent (e.g. `q.PP|ADJP|VP(x0:IN x1:NP|PP|ADJP)` $\rightarrow$ `A q.x1`). Performance of all these models on the dev set is reported in Table 2 for accents and in Table 3 for breaks. Finally, the best models from the development phase were evaluated on the test set, which consisted entirely of unseen sentences. These results are shown in Table 4 for accents and Table 5 for breaks.

5. Discussion

Overall, the tree transducer models outperformed the n-gram baselines. On the dev set this was about 12% absolute improvement for accents and about 5% for breaks, while on the test set it was 14% for accents and 3% for breaks. Though the n-gram models had a 14% higher break detection accuracy on the test set, they performed worse overall when false insertions were factored in. The tree transducer's superiority in overall performance is statistically significant: in both accent and break prediction, McNemar's test showed the effect of the tree transducer models to be different than the effect of the baseline models ($p < 0.01$). On both accents and breaks, only the tree transducer results fell within one standard deviation of the average inter-speaker agreement as to how the sentences should be produced (as reported in Section 2). In other words, the tree transducer placed accents and breaks as accurately as a newscaster would.

The best tree transducer model on the dev set was the one that incorporated the deepest tree context: the grandparent-dependent (GD) models. The best baseline models on the dev set used 2nd-order n-grams, in contrast to the experiments in [10] which concluded a 6th-order model to be best. There are a number of differences between our baseline implementation and the original in [10] (not the least of which being the significantly larger train set in [10]). However, on a subset of BURNC data (though not the present study's specific test set) [10] reported a 72.72% break accuracy - this is close to the 74.78% reported here in Table 5. To be fair, the work in [10] only examined break (and not accent) prediction - perhaps this baseline is not ideal for accents.

In analyzing the errors made by the best model - the GD tree transducer - we see that it is weak in predicting accents for most types of verb tags, and also for nouns. Predicting breaks after nouns was also less reliable than for most other POS tags. This suggests that the tree transducer methods would benefit from clustering the POS tags into a smaller set - combining all noun tags into one, all verb tags into one, etc. - as done in [10].

6. Conclusion

The connection between high-level syntax and categorical prosody has been known for a long time. This study has shown that it is sensible to model this connection using tree transducers - they are capable of automatically determining where accents and breaks should fall in unseen text more accurately than a standard baseline model, and with accuracy comparable to the average inter-speaker agreement in determining the same. Tree transducers allow for much creativity and flexibility - recall the improvement seen in transforming these models into parent- and grandparent-dependent versions. Future work in this area will benefit from modeling deeper and wider syntactic contexts in the transduction rules. And read newscaster speech is only the beginning - it will be interesting to see how this method performs on spontaneous or nonnative speech.

7. References

[1] J. Bachenko and E. Fitzpatrick, "A Computational Grammar of Discourse-Neutral Prosodic Phrasing in English," *Computational Linguistics*, 16:155-170, 1990.

[2] D. Bolinger, "Accent is Predictable (if You are a Mind-Reader)," *Language*, 48:633-644, 1972.

[3] N. Chomsky and M. Halle, *The Sound Pattern of English*, MIT Press, 1968.

[4] J. Hirschberg and O. Rambow, "Learning Prosodic Features Using a Tree Representation," in *Proc. of Eurospeech*, Aalborg, 2001.

[5] D. Klein and C.D. Manning, "Accurate Unlexicalized Parsing," *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 2003.

[6] J. May and K. Knight, "Tiburon: A Weighted Tree Automata Toolkit," in *Proc. of the Eleventh Conference on Implementation and Application of Automata*, 2006.

[7] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University Radio News Corpus," Boston University Technical Report No. ECS-95-001, March 1995.

[8] J. Pierrehumbert, *The phonology and phonetics of English intonation*, PhD thesis, MIT, 1980.

[9] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A Standard for Labeling English Prosody," in *Proc. of ICSLP*, Banff, Canada, 1992.

[10] P. Taylor and A.W. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, 12:99-117, 1998.

[11] J. Tepperman and S. Narayanan, "Tree Grammars as Models of Prosodic Structure," in *Proceedings of InterSpeech ICSLP*, 2008.