A Hierarchical Stochastic Model for Automatic Prediction of Prosodic Boundary Location

M. Ostendorf*
Boston University

N. Veilleux*
Boston University

Prosodic phrase structure provides important information for the understanding and naturalness of synthetic speech, and a good model of prosodic phrases has applications in both speech synthesis and speech understanding. This work describes a statistical model of an embedded hierarchy of prosodic phrase structure, motivated by results in linguistic theory. Each level of the hierarchy is modeled as a sequence of subunits at the next level, with the lowest level of the hierarchy representing factors such as syntactic branching and prosodic constituent length using a binary tree classification. A maximum likelihood solution for parameter estimation is presented, allowing automatic training of different speaking styles. For predicting prosodic phrase breaks from text, a dynamic programming algorithm is given for finding the maximum probability prosodic parse. Experimental results on a corpus of radio news demonstrate a high rate of success for predicting major and minor phrase boundaries from text without syntactic information (81% correct prediction with 4% false prediction).

1. Introduction

Prosodic phrase structure plays a role in both naturalness and intelligibility of speech. For example, prosodic phrase boundaries break the flow of a sentence, dividing it into smaller units for easier processing. In addition, researchers have shown that prosodic phrase break placement is important in syntactic disambiguation (Lehiste 1973; Price, Ostendorf, Shattuck-Hufnagel, and Fong 1991). For these reasons, computational modeling of prosodic phrases is important both for text-to-speech synthesis and speech understanding applications. In this work, we present a computational model that represents a hierarchy of prosodic constituents using a stochastic formalism to capture the natural variability allowable in prosodic phrasing. The model is useful for both analysis and synthesis applications; we focus on synthesis here, and present experimental results for predicting prosodic phrase structure from text.

Prosodic phrase structure, or groupings of words in a sentence, can be equivalently represented by different phrase break markers. The location and relative size of these breaks define the prosodic phrase structure, which we will refer to here as a prosodic parse. Prosodic phrase breaks are discrete events that are associated with acoustic cues such as duration lengthening, pause insertion, and intonation markers. In this work, we are concerned only with the relationship between the abstract events

* ECS Department, 44 Cummington Street, Boston, MA 02215

(different levels of phrase breaks) and text. To be useful in synthesis or understanding applications, the results presented here need to be integrated with a component that models the acoustics associated with these abstract events (see, for example, Hirose and Fujisaki 1982).

Several observations about prosodic phrase breaks raise issues to be considered in designing an algorithm to predict such breaks from text. First, there is a significant body of literature in linguistics concerning various hierarchies that specify the relationship among prosodic constituents, and the model should reflect this structure. Second, several different prosodic parses may all be acceptable for one sentence. This variability is particularly important to represent if the model is to be useful for analysis as well as synthesis. Third, prosodic phrase breaks do not always coincide with syntactic phrase boundaries, and the relationship between prosody and syntax is not well understood. This means that prosodic phrases cannot simply be predicted from syntactic structure. Finally, since most text-to-speech synthesis applications require a low cost implementation, there is the concern of computational complexity. We shall expand on these points separately below, to motivate the work described here.

The various linguistic theories of prosodic phrase structure (e.g., Liberman and Prince 1977; Selkirk 1980, 1984; Beckman and Pierrehumbert 1986; Nespor and Vogel 1983; Ladd 1986) differ in the specific levels that they represent, but all have a similar hierarchical structure. Two levels of prosodic phrases are common to most proposals: the intonational phrase and the intermediate phrase, using the terminology of Beckman and Pierrehumbert. A sentence is composed of a sequence of intonational phrases, which in turn are composed of sequences of intermediate phrases. An intonational phrase break is therefore perceived as stronger or more salient than an intermediate phrase break. Intonational phrases are delimited by boundary tones, and intermediate phrases are theoretically marked with a phrase accent, where the pitch markers can be either high or low (Beckman and Pierrehumbert 1986). (In other theories of intonation, for example, t'Hart, Collier, and Cohen [1990], pitch markers also occur at phrase boundaries, but are identified with movement and referred to as either rising or falling.) Both types of constituents are also cued by segmental lengthening in the phrase final syllable (Wightman, Shattuck-Hufnagel, Ostendorf, and Price 1992). Since intonational and intermediate phrases are generally accepted, the experiments here will only address these two levels, referring to them as major and minor phrases, respectively. However, other types of prosodic constituents may be useful and, in fact, there is durational evidence for at least four levels (Wightman, Shattuck-Hufnagel, Ostendorf, and Price 1991; Ladd and Campbell 1991). We therefore propose a more general hierarchical model that can be extended to an arbitrary, but fixed, number of levels.

In the examples given here, we will represent intonational phrases (I) using "||" to mark a major break and intermediate phrases (i) using "|" to mark a minor break. The example below illustrates how phrase breaks are used to represent prosodic phrase structure:

Those on early release | *must check in with correction officials* ||
fifty times a week || *according to Ash,* ||
who says about half | *the contacts for a select group* ||
will now be made | *by the computerized phone calls.* ||

((*Those on early release*)$_i$ (*must check in with correction officials*)$_i$)$_I$
((*fifty times a week*)$_i$)$_I$ ((*according to Ash,*)$_i$)$_I$
((*who says about half*)$_i$ (*the contacts for a select group*)$_i$)$_I$
((*will now be made*)$_i$ (*by the computerized phone calls.*)$_i$)$_I$

Another important consideration in modeling prosody (and evaluating the model) is that prosodic phrase structure is not deterministic. Speakers can produce a sentence in several ways without altering the naturalness or the meaning. Prosodic breaks can differ in size and/or placement because of differences in style, competence, or simply natural speaking variations. For example, the following sentence was said three ways by five speakers:

They're in jail || for such things || as bad checks or stealing.
They're in jail | for such things | as bad checks | or stealing.
They're in jail || for such things as bad checks | or stealing.

Although deterministic rules can be used to predict phrase breaks for speech synthesis applications, such a model will be limited in its usefulness in speech analysis. In addition, speech synthesis might be more natural if variability is included in the model. Here, a stochastic model is used to represent the natural variability in prosodic structure by deriving probabilities of phrase breaks, rather than predicting locations of phrase breaks by rule.

The relationship between prosody and syntax is not fully understood, though it is generally accepted that there is such a relationship. For example, relatively higher syntactic attachment usually corresponds to relatively larger prosodic breaks, but there are many exceptions, as in:

[[Mary]np [was amazed [Ann Dewey was angry]s']vp]s

which was produced by four speakers as

Mary was amazed || Ann Dewey was angry.

In an analysis of the London–Lund corpus, Altenberg (1987) finds relative frequencies that describe the correspondence between prosodic constituents (tone units) and different syntactic units. This data supports the use of a probabilistic model, which also has an advantage in that it can be trained automatically, facilitating representation of a wide variety of speaking styles and allowing a means of discovering syntax-prosody relationships from a large corpus. One reason that the mapping between syntax and prosody is not simple is because, in speech, the constraints of syntactic structure and phrase length are balanced to produce a regular, roughly equal, sequence of prosodic phrases (Gee and Grosjean 1983). Consequently, we include constituent length as a factor in the model.

The cost of obtaining a full and accurate syntactic parse can be high, which presents difficulties for text-to-speech synthesis systems. In addition, a full syntactic parse may not be necessary for predicting prosodic phrases, since prosody is not directly related to syntax. Consequently, we investigate computation/performance trade-offs associated with using a skeletal syntactic parse vs. simple part-of-speech (POS) assignments.

To summarize, the model proposed here addresses several issues in modeling prosodic phrase structure. The model is a general formalism for an embedded hierarchy, which we specifically apply to represent sentences, major phrases, and minor phrases. In order to account for the allowable variability in prosodic parsing, the model is probabilistic. The structure of the model allows use of grammatical information such as part-of-speech labels, syntactic structure and constituent length, but the specific parameters are trained automatically. Finally, computational complexity trade-offs are investigated by evaluating the algorithm with and without syntactic cues.

The remainder of the paper is organized as follows. We begin, in Section 2, by discussing past work in predicting prosodic phrase breaks from text for speech synthesis. In Section 3, we introduce the probabilistic formalism of the hierarchical model and outline the implementation: text pre-processing, parameter estimation, and phrase break prediction using a dynamic programming algorithm to obtain the most likely prosodic parse. In Section 4, we present experimental results for prediction of major and minor prosodic phrase breaks based on a corpus of FM radio news stories. Finally, we conclude in Section 5 by discussing possible implications and extensions of these results.

2. Previous Work

Initial attempts to incorporate prosody in speech synthesis involved determining intonation and duration patterns as a function of syntactic phrase structure (Allen, Hunnicutt, Carlson, and Granstrom 1979; Allen, Hunnicutt, and Klatt 1987), which requires syntactic parsing. More recently, researchers have attempted to address the fact that prosody and syntax are not directly related by explicitly predicting prosodic phrase boundaries rather than using syntactic clause boundaries. An important difference between these subsequent approaches is in the amount of syntactic information used to predict prosodic boundaries. The algorithms reflect different assumptions about the relationship between prosody and syntax, as well as different levels of computational complexity. Clearly, a greater use of syntactic information will require more computation for finding a more detailed syntactic parse.

One approach is based on the idea that a prosodic parse may not require a full syntactic parse and that detailed part-of-speech information (e.g., noun, verb, determiner) may not be necessary for generating a prosodic parse. Sorin, Larreur, and Llorca (1987) proposed a simple prosodic parser for French based on content/function word classification to determine prosodic constituents referred to as prosodic groups. The length and relative location of these prosodic groups is then used to determine phrase break locations that are marked with a pause. Our earlier work drew on this scheme for predicting phrase boundaries in English: a Markov model was developed to predict phrase breaks by representing the sequence of prosodic groups and breaks as a Markov chain (Veilleux, Ostendorf, Price, and Shattuck-Hufnagel 1990). An advantage of these approaches is that they only require a small dictionary of function words to assign part-of-speech labels. Motivated by similar principles and using only a 300-word dictionary, O'Shaughnessy (1989) proposes a somewhat more sophisticated parser for English based on function word identification, number agreement, and suffix identification. O'Shaughnessy's work differs from the other approaches in that his goal is a syntactic parse, though not complete, and he does not address the issue of differences between prosody and syntax.

At the other end of the spectrum are approaches based on the hypothesis that prosodic phrase boundaries can be predicted by rule from a full syntactic parse. Gee and Grosjean (1983) developed a rule-based system, called the **Phi Algorithm**, to predict psycholinguistic "performance structures" that are represented by assigning an integer number corresponding to boundary salience between each pair of words. Constituent length information is incorporated primarily through the application of their verb balancing rule, which splits the verb phrase and groups the verb with either the previous or subsequent material, subject to syntactic constraints. Gee and Grosjean developed their Phi Algorithm only to predict performance structures. However, their work has been extended to prosodic phrase prediction for speech synthesis applications by Bachenko and Fitzpatrick (1990), who explicitly find prosodic phrase breaks from derived boundary salience indices. They relax many of the constraints on the use of the verb rule and propose a Verb Adjacency Rule, so their algorithm requires a fairly detailed parse, although not a complete one. One of the relaxed constraints obviates the need for clause information. Altenberg (1987) has also proposed an algorithm for prediction of phrase boundary locations (specifically, tone unit boundaries for British English) by rule from syntactic structure and semantic information. However, the detailed information required for the algorithm cannot currently be acquired automatically from text.

Departing from these approaches, Wang and Hirschberg (1992) have recently used binary decision trees to predict the presence or absence of a prosodic break at each word boundary in a sentence. They consider a range of input variables, including text-derived information such as detailed POS labels and syntactic constituent structure, and in some experiments, acoustic information. POS labels were given by Church's tagger (Church 1988) and syntactic constituents by Hindle's parser (Hindle 1987). The acoustic information (previous boundary location, pitch accent location, and phrase duration), which was based on hand-labeled prosodic markers, did not improve performance but resulted in a much smaller tree for prediction.

All of these approaches have influenced the model proposed here. For example, we investigate simple content/function word POS assignment, as in Sorin, Larreur, and Llorca (1987). Like Wang and Hirschberg (1992), we use decision trees to automatically determine the important factors influencing phrase break location. In addition, all of the above works have influenced the choice of factors and questions incorporated in the decision tree. Two important differences in our approach include a stochastic model to capture variability and an explicit representation of a linguistically motivated hierarchy. Of course, whether it is effective and/or efficient for a computational model to reflect a linguistic hierarchy is an empirical question.

3. Hierarchical Model of Prosodic Phrases

A prosodic parse of a sentence can be represented by a sequence of break indices, one index following each word, which code the level of bracketing or attachment in a tree. A prosodic parse S is therefore given by

$$S = (b_1, b_2, \ldots, b_L),$$

where b_i is the break index after the ith word and L is the number of words in the sentence. A **break** is a random variable that can take on one of a finite number of values from "no break" (orthographic word boundary, but not a prosodic constituent boundary) to "sentence boundary," where the values form an ordered set that correspond to the different levels of the hierarchy. Below we consider a stochastic model for first a general hierarchical prosodic parse (any specified number of levels), and then specifically for the three-level case that models a sentence as a sequence of **major phrases,** which are in turn modeled as a sequence of **minor phrases.** Although most phonological theories do not recognize the "sentence" as a unit, it is useful for both synthesis and recognition applications to model sentences separately, as sentence-final boundaries tend to be acoustically different from sentence-internal boundaries (e.g., a low boundary tone is much more likely).[1]

[1] We have chosen to use the term "sentence" rather than the more general term "utterance," since the algorithm is designed to predict boundaries from text that in our data and in many applications

We will begin by presenting the mathematical structure, first generally and then specifically as a three-level embedded hierarchy. Next, some pragmatic details of text processing are discussed, followed by a description of the parameter estimation and phrase break prediction algorithms.

3.1 Stochastic Model

We assume the relationship between units and subunits will hold at any level of the hierarchy. Therefore, in describing the general case, we need only consider one level of embedding and will use U_i and u_{ij} when referring to units and subunits, respectively, at some unspecified level of the hierarchy. Using this notation, the probability of a unit U_i is parameterized in terms of the probability of the sequence of subunits u_{ij} (on the next lower level) and the length n_i in subunits of that sequence given the orthographic transcription of the sentence $\mathcal{W}$:

$$\begin{aligned} \mathrm{p}(U_i|\mathcal{W}) &= \mathrm{p}(u_{i1},\ldots,u_{in_i}|\mathcal{W}) \\ &= \mathrm{p}(u_{i1},\ldots,u_{in_i}|\mathcal{W},n_i)\mathrm{p}(n_i|\mathcal{W}) \\ &= \mathrm{p}(n_i|\mathcal{W})\mathrm{p}(u_{i1}|\mathcal{W})\prod_{j=2}^{n_i}\mathrm{p}(u_{ij}|\mathcal{W},u_{i1},\ldots,u_{i(j-1)}). \end{aligned}$$

The specific hierarchy considered here involves representing the prosodic parse of a sentence S as an N-length sequence of major phrases M_i:

$$S = (M_1,\ldots,M_N).$$

A major phrase M_i is composed of a n_i-length sequence of minor phrases m_{ij}:

$$M_i = (m_{i1},\ldots,m_{in_i}).$$

Finally, a minor phrase m_{ij} is composed of a ν_{ij}-length sequence of breaks b_t starting at time $t(i,j)$ and ending at time $t(i,j+1)-1$,

$$m_{ij} = (b_{t(i,j)},\ldots,b_{t(i,j+1)-1}),$$

where $t(i,j)$ is the time index of the first word of m_{ij} and $t(i,j+1)-1$ is the time index of the final word of m_{ij}, $\nu_{ij} = t(i,j+1) - t(i,j)$ is simply the number of words in the minor phrase, and the breaks b_t take on values from the set {no break, minor break, major break, sentence break}.

It might be useful to consider phonological words rather than orthographic words as possible sites for break indices. This could be accomplished, without using deterministic rules, by specifying the bottom of the hierarchy (e.g., break level 0) to represent locations internal to a phonological word and the next level of the hierarchy (e.g., break level 1) to represent phonological word boundaries. However, it is controversial as to whether phonological words can be larger or smaller than orthographic (lexical) words (Booij 1983; Nespor and Vogel 1983), so it is not clear how the lowest level should be defined relative to the orthographic words. In this work, we have chosen not to distinguish between these two levels, to reduce the complexity of implementation and performance evaluation. For similar reasons, we have limited

comprises syntactically well-formed sentences. The phrase prediction model may also be useful in speech recognition applications, in which case the term "utterance" would clearly be more appropriate.

Table 1
Histograms of number of minor phrases in a major phrase and number of major phrases in a sentence, as a function of quantized length of the unit. The quantizer regions are indicated by the length ranges.

Number of minor phrases in major phrase					
$\ell(M)$	1	2	3	4	5
1–5	522	69	2		
6–7	106	78	9		
8–14	61	107	43	1	
15–19		2	2	1	1

Number of major phrases in sentence						
$\ell(S)$	1	2	3	4	5	6
1–14	31	39	11	2	1	
15–21	4	23	28	17	9	5
22–31		6	6	20	24	11
32–37				2	6	4

this study to the more universally agreed-upon levels of major and minor prosodic phrases, although there is durational evidence that a more detailed hierarchy would be useful (Ladd and Campbell 1991; Wightman, Shattuck-Hufnagel, Ostendorf, and Price 1992).

In the current work, we make several simplifying assumptions due to training limitations. First, the probability of the number of subunits in a unit $p(n_i|U_i)$ is assumed to depend only on the number of words in the unit $\ell(U_i)$. As is not surprising, our data indicate that units that span a larger number of words tend to comprise more subunits. Altenberg has noticed similar tendencies in the London–Lund corpus (Altenberg 1987, p. 81). (Alternatively, it has been suggested that either phonological word count or stressed syllable count rather than orthographic word count may be a useful measure of phrase length on the lowest level [Bachenko and Fitzpatrick 1990].) In addition, the probability distribution is approximated by conditioning on quantized lengths $Q_U(\ell(U_i))$. The quantizer varies as a function of the specific unit and is designed using a regression tree (Breiman, Friedman, Olshen, and Stone 1984). A regression tree partitions the data along intervals of a continuous variable, in this case length of the unit, to decrease variance of the response variable, the number of subunits in the unit. The resulting quantizer regions and the corresponding distribution of subunits in a unit are given in Table 1 for major phrase and sentence units.

Using these simplifying assumptions, the constituent length probability distributions are then:

$$\mathrm{p}(N|\mathcal{W}) = q_S(N|L) \quad (1)$$
$$\mathrm{p}(n_i|\mathcal{W}) = q_M(n_i|l_i) \quad (2)$$
$$\mathrm{p}(\nu_{ij}|\mathcal{W}) = q_m(\nu_{ij}|\lambda_{ij}) \quad (3)$$

where $L = Q_S(\ell(S))$, $l_i = Q_M(\ell(M_i))$ and $\lambda_{ij} = Q_m(\ell(m_{ij}))$.

Next, major phrases in a sequence are assumed to be Markov given the number in the sequence:

$$\mathrm{p}(M_i|\mathcal{W}, M_1, \ldots, M_{i-1}) = p(M_i|\mathcal{W}, M_{i-1}).$$

Minor phrases are also assumed to be Markov, depending only on the previous minor phrase and the features of the major phrase it is contained in:

$$\mathrm{p}(m_{ij}|\mathcal{W}_i, m_{i1}, \ldots, m_{i(j-1)}, M_{i-1}) = p(m_{ij}|\mathcal{W}_i, m_{i(j-1)})$$
$$\mathrm{p}(m_{i1}|\mathcal{W}_i, M_{i-1}) = p(m_{i1}|\mathcal{W}_i, m_{(i-1)n_{i-1}}),$$

where $\mathcal{W}_i$ is the sequence of feature vectors spanning the ith major phrase. For simplicity, we will abbreviate the notation to:

$$\mathrm{p}(m_{ij}|\mathcal{W}_i, m_{i1}, \ldots, m_{i(j-1)}, M_{i-1}) \quad = \quad p(m_{ij}|\mathcal{W}_i, m_{prev}).$$

The conditional probability of a sequence of words within a minor phrase is assumed to depend on a state determined by the (variable-length) sequence of past words and the time of the last break, where the state is given by a decision tree as in Bahl, Brown, deSouza, and Mercer (1989):

$$\mathrm{p}(b_k|\mathcal{W}_i, b_{t(i,j)}, \ldots, b_{k-1}, m_{prev}) = \mathrm{p}(b_k|f(\mathcal{W}_i, m_{prev})).$$

Note that within a minor phrase, probabilities for only two cases, that of no break or that of any higher level break index, are used.

Incorporating all of the above simplifying assumptions, the probability of a specific prosodic parse is given by

$$\mathrm{p}(S|\mathcal{W}) \quad = \quad q_S(N|L) \prod_{i=1}^{N} \mathrm{p}(M_i|\mathcal{W}, M_{i-1}), \tag{4}$$

$$\mathrm{p}(M_i|\mathcal{W}, M_{i-1}) \quad = \quad q_M(n_i|l_i) \prod_{j=1}^{n_i} \mathrm{p}(m_{ij}|\mathcal{W}_i, m_{prev}), \tag{5}$$

$$\mathrm{p}(m_{ij}|\mathcal{W}_i, m_{prev}) \quad = \quad q_m(\nu_{ij}|\lambda_{ij}) \prod_{k=t(i,j)}^{t(i,j)+\nu_{ij}-1} \mathrm{p}(b_k|f(\mathcal{W}_i, m_{prev})). \tag{6}$$

$\mathcal{W}$ used in this model is not simply the orthographic word sequence. Rather, it is a sequence of feature vectors, one per word extracted from the word sequence. Examples of possible features include part-of-speech labels and syntactic information such as bracketing labels or labels of an associated node in a syntactic tree. The decision tree $f(\mathcal{W}_i, m_{prev})$ used in determining the probabilities $\mathrm{p}(b_k|f(\mathcal{W}_i, m_{prev}))$ includes questions based on these features, attributes of the previous minor phrase and the current major phrase, and length in words of the sentence. Details on our specific choice of features and questions is given in Section 4.

Our use of decision trees is different from the phrase break detection algorithm of Wang and Hirschberg (1992), although the tree design algorithm and choice of features is similar. The tree is not used to classify phrase breaks directly; instead it is used to determine the probability of the occurrence of a minor break at some location, conditioned on the decision tree structure. This probability is used to represent the lowest level in the hierarchical model.

Previously, we mentioned two important factors affecting the placement of phrase breaks: (1) grammatical structure and (2) length constraints on the prosodic constituents such as overall length and length relative to neighboring phrases. Grammatical information is incorporated in the tree $f(\mathcal{W}_i, m_{prev})$ through questions about the feature sequence $\mathcal{W}_i$. Prosodic constituent length is modeled in two ways, through the constituent length probability distributions and through questions about the length of the previous phrase used in the tree $f(\mathcal{W}_i, m_{prev})$.

3.2 Text Processing

In the experiments reported here, the feature vectors include part-of-speech labels, punctuation and, optionally, information from a skeletal syntactic parse. The feature extraction is described in more detail below, and an example is given in Figure 1.

Two levels of detail are considered for part-of-speech (POS) labeling. At the simplest level, a function word table look-up is used to categorize words either as one of

	[VP is	[AP free	[PP on	[NP bail]]	[PP after	[NP [VP facing . . .
word # in sent.	5	6	7	8	9	10
word class	v	c	p	c	p	c
part-of-speech	v	adj	p	noun	p	verb
punctuation	none	none	none	none	none	none
left dominated	same	same	same	PP	same	—
right dominated	AP	PP	NP	PP	NP	—
both dominated	VP	AP	PP	AP	PP	—
# init. constit.	1	1	1	1	2	—
# term. constit.	0	0	0	2	0	—

Figure 1
Seven features are extracted for each word in a sentence to describe the boundary between that word and the following word. Syntactic information is based on a skeletal parse, as shown. Part-of-speech assignment is based on a table look-up from lists of function words.

six types of function words, as a proper name (P) if capitalized, or otherwise as a content word (c). Function words are divided into several classes: conjunctions (j) (such as *and, but, if, because*), auxiliary verbs and modals (v), determiners (d), prepositions (p), pronouns (n), and a general category (g), which includes the quantifiers and function-like adverbs such as *not, no, ever, now*. The POS labels given by the simple table look-up are referred to here as "word classes." A more detailed part-of-speech classification is given by Penn Treebank POS tags (Marcus and Santorini 1993), which were obtained automatically using the BBN tagger (Meteer, Schwartz, and Weischedel 1991). What we refer to here as POS labels is actually a grouping of these classes that includes the above function word categories, the proper name category (now determined by the tagger rather than from capitalization), plus categories for particles (pa), nouns (noun), verbs (verb), adjectives (adj), adverbs (adv), and all other content words (def).

Contractions are not decomposed into separate words, since it is not possible that a phrase break will occur within the word contraction. The contraction is treated as a single word in constituent length measures and feature extraction, and it is assigned the POS label of its base word (left component).

Punctuation following a word is incorporated as a feature for that word. In our data, the only punctuation that appears are commas and periods. Periods and other sentence-final punctuation deterministically assign a sentence break. This implies some text preprocessing that distinguishes periods used for abbreviation from sentence-final periods. While commas often correspond to major breaks, there is a systematic exception: a series of the same syntactic units such as a series of nouns (*an apple, an orange, and a pear*) or a series of adjectives (*safe, cost-effective alternative ...*) may or may not be associated with a major prosodic break. Therefore, we have chosen to use commas as a feature to determine the likelihood of a phrase break rather than as a deterministic cue to a prosodic break. Although using commas deterministically to assign a major phrase boundary yields better performance on our test set than using commas as a feature, we felt that using commas as a feature was a more extensible approach, and have used this strategy in the results reported here. Including commas as a feature does improve performance relative to not using commas, as will be discussed in Section 4. Commas (and other punctuation) can be very useful for prosodic boundary prediction when they are available, and they are used in other algorithms (e.g., Allen, Hunnicutt, and Klatt 1987; O'Shaughnessy 1989; Bachenko and Fitzpatrick 1990). However, commas are not reliably transcribed from spoken language and not consistently used in written text, so it is important that the algorithm not depend too heavily on commas.

Syntactic features were extracted from skeletal parses provided through a preliminary version of the Penn Treebank Corpus. Since these are hand-corrected parses, the results are indicative of the performance possible using syntactic information, but do not reflect performance achievable with an existing parser. Several researchers have investigated the relationship between prosody and syntax (e.g., Selkirk 1984; Gee and Grosjean 1983; Cooper and Paccia-Cooper 1980; Altenberg 1987). Our features have been motivated by some of these results, which suggest that some syntactic constituents are more likely to be separated by a phrase break than others. However, we have chosen to let the important constituents be determined automatically, similar to Wang and Hirschberg (1992), rather than by rule. One feature is the highest syntactic constituent dominating the left word but not dominating the right word, which describes potential locations for phrase breaks after a specific syntactic constituent. We also consider the similar case, the highest syntactic constituent dominating the right word but not the left, to allow for prosodic phrase breaks that may be associated with the beginning of a syntactic constituent. The lowest syntactic constituent that dominates both words is a feature that will provide information about which constituents are not likely to be divided by a phrase break. In addition, the number of terminating constituents and the number of initiating constituents between the two words were included as features to investigate the influence of relative strength of syntactic attachment. Eight categories of syntactic constituent were used: sentence (S), noun phrase (NP), verb phrase (VP), prepositional phrase (PP), wh-noun phrase (WHNP), adjective or adverbial phrase (AP), any other constituent (O), and both words in the same lowest level constituent (same).

3.3 Parameter Estimation

An advantage of a stochastic model is that the parameters can be estimated automatically from a large corpus of data, which means that it is relatively straightforward to redesign the model to reflect a different speaking style. Here we describe a maximum likelihood approach to parameter estimation, where model parameters are chosen to maximize the likelihood of the training data.

We will assume that sentences are independent and identically distributed to simplify parameter estimation and prediction, although the independence assumption precludes capturing any speaker-dependent or discourse effects. In this case, the likelihood of the prosodic parse of a corpus of sentences $(S^1, \dots, S^T)$ given parameters θ is, from Equations (4)–(6),

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_t \log \mathrm{p}(S^t|\mathcal{W}^t) \\
&= \sum_t \left[\log q_S(N^t|L^t) + \sum_i \log \mathrm{p}(M_i^t|\mathcal{W}^t, M_{i-1}^t) \right] \\
&= \sum_t \left[\log q_S(N^t|L^t) + \left[\sum_i \left[\log q_M(n_i^t|l_i^t) + \sum_j \log \mathrm{p}(m_{ij}^t|\mathcal{W}_i^t, m_{prev}^t) \right]\right]\right] \\
&= \sum_t \Bigg[\log q_S(N^t|L^t) + \Bigg[\sum_i [\log q_M(n_i^t|l_i^t) + \\
&\qquad \left[\sum_j \left[\log q_m(\nu_{ij}^t|\lambda_{ij}^t) + \sum_k \log \mathrm{p}(b_k^t|f(\mathcal{W}_i^t, m_{prev}^t)) \right]\right]\Bigg]\Bigg] .
\end{aligned}$$

Arranging terms we have

$$
\begin{aligned}
\mathcal{L}(\theta) &= \sum_t \log q_S(N^t|L^t) \\
&+ \sum_t \sum_i \log q_M(n_i^t|l_i^t) \\
&+ \sum_t \sum_i \sum_j \log q_m(\nu_{ij}^t|\lambda_{ij}^t) \\
&+ \sum_t \sum_i \sum_j \sum_k \log \mathrm{p}(b_k^t|f(\mathcal{W}_i^t, m_{prev}^t)). \qquad (7)
\end{aligned}
$$

Since there are no cross dependencies between parameters, the four terms in Equation (7) can be maximized separately. The resulting parameter estimates for q_S, q_M, and q_m are then simply relative frequency estimates. The last term is maximized jointly with the design of the state function $f(\cdot)$ using standard classification tree design techniques, as described below.

The tree is grown using a greedy algorithm, which iteratively extends branches by choosing the parameters of a question, the question at a node and the node in a tree that together maximize some criterion for reducing the impurity of the class distributions at the leaves of the tree. The tree is used to find the probability of a minor phrase boundary, so there are only two classes: "break" and "no break." In this work, we have used the Gini criterion, i.e., the node distribution impurity is given by $i(t) = \sum_{i \neq j} p(i|t)p(j|t)$ (Breiman, Friedman, Olshen, and Stone 1984). Since the relative frequency of the "break" class is so low (8% of all breaks), we include different error costs in the design criterion. Generally, the cost of classifying a "break" as "no break" is chosen to be three to four times higher than the opposite error, and the specific costs for each tree are chosen to control the false prediction rate on the training set. Initially a tree is grown using two-thirds of the training data, and the remaining one-third of the data is used to determine a good complexity-performance trade-off point. The complexity criterion determined at this point is then used in pruning a second tree grown with the entire training set, in order to make better use of the available data. Each leaf $\tilde{t}$ of the tree is associated with a conditional probability distribution of "break" vs. "no break" (actually, the relative frequency estimate). This probability distribution $p(b|\tilde{t})$ is used in the hierarchical model for computing the probability of a minor phrase, Equation (6), by running test data through the tree and using the probability distribution associated with the final leaf node $\tilde{t} = f(\mathcal{W}_i, m_{prev})$.

3.4 Phrase Break Prediction Algorithm

The stochastic model can be used to predict a prosodic parse for a sentence simply by finding the most probable prosodic parse for that sequence of words, where the probability of any given parse is determined by Equations (4)–(6). In other words, we hypothesize all possible prosodic parses, compute the probability of each, and choose the most probable. The most likely prosodic parse can be found efficiently using a dynamic programming (DP) algorithm that is similar to algorithms used in speech recognition, in particular that for the Stochastic Segment Model (Ostendorf and Roukos 1989), except that the dynamic programming routine is called recursively for successive levels in the hierarchy. Defining $p_t(u_{i1}...u_{in}|\mathcal{W}_i, U_i)$ as the probability of the most likely sequence of n subunits in but not necessarily spanning U_i and ending at location t, and $u_{ij}(s,t)$ as a subunit that spans boundaries $\{b_s, \ldots, b_t\}$, the dynamic programming algorithm can be expressed generally in the subroutine that follows. This subroutine is called recursively for each level of the hierarchy, with the lowest level constituent probability being computed using probabilities given by the tree.

Dynamic Programming Routine for Prosodic Parse Prediction

```
For each word $t$ in unit $U_i$ $(t = 1, \ldots, l_i)$:
  Compute $\log p_t(u_{i1}(1,t) | W_i, U_{i-1})$.
  For each $n$-length sequence of subunits spanning $[1,t]$ $(n = 2, \ldots, t)$:
      $\log p_t(u_{i1} \ldots u_{in} | W_i, U_{i-1}) = \max_{s<t} \log p_s(u_{i1}, \ldots, u_{i,(n-1)} | W_i, U_{i-1})$
                                      $+ \log p(u_{in}(s+1,t) | W_i, u_{i(n-1)})$
      (Computing $\log p(u_{in}(s+1,t) | W_i, u_{i(n-1)})$ with a recursive call to this routine.)
      Save pointers to best previous break location $s$.
To find the most likely sequence,
  $p(U_i | W_i, U_{i-1}) = \max_n \log p_{l_i}(u_{i1}, \ldots, u_{in} | W_i, U_{i-1}) + \log q(n | l_i)$
```

The final step is to decode the sequence of breaks once the value n^* that maximizes the above equation is determined. Using the n^* associated with any level unit, we can trace back to find the optimal segmentation of subunits that comprise that unit. The complete parse is found by tracing back at the highest level units and successively tracing back in each lower level.

For the specific case of a three-level hierarchy, the most likely major phrase sequence in a sentence p$(S|W)$ and the most likely minor phrase sequence in a major phrase p$(M_i|W)$ are found by a dynamic programming algorithm, called recursively. The lowest level unit considered here is the minor phrase, and the probability of the minor phrase is computed as given in Equation (6) using the decision tree.

4. Experiments

4.1 Corpus

For our investigation of prosodic phrase structure, an FM radio news story corpus was used. The training data included ten stories from one announcer and another ten stories from a second announcer, both female, for a total of 312 sentences (6,157 words, or potential boundary locations). The stories were studio recordings of actual radio broadcasts, which were transcribed by a listener who did not have access to the original scripts. It is likely that transcription of punctuation did not exactly match the original written text and may have been biased by the prosody of the utterance. However, the radio announcers tended to annotate the transcribed text before reading the test stories, so we conjecture that commas were more often omitted than inserted in our transcriptions. All of the training stories were used to estimate the probabilities of the number of subconstituents (Equations 1–3). In the first pass of tree design, two-thirds of the training data was used to grow the tree and one-third was used to determine the performance complexity trade-off, but the final tree used was redesigned on the entire training set.

For testing, we used five versions of a different story spoken by two female and two male announcers (one radio broadcast version and four radio-news-style lab recordings). One of the female announcers (two spoken versions) was the same as the speaker who provided roughly three-quarters of the training data. Multiple test versions are used in order to allow for some acceptable differences in phrasing in the context of the FM radio news style, and to investigate the possibility of speaker-dependent effects. On average, there were 3.3 different prosodic parses among the five versions. The test story contained 23 sentences (385 words) ranging in length from 3 to 36 words. For reference, the test sentences are included in an appendix with the phrase predictions of our best system.

Prosodic phrase breaks were hand-labeled in the entire corpus; the training set labels were used for estimating the parameters of the model and test set labels were used for evaluating the performance of the model. The prosodic phrase labeling system used break indices marked between each pair of words, based on auditory perceptual judgments (that is, the labelers did not have access to spectrogram or pitch displays). The break indices ranged on a scale of 0 to 6, chosen to map to a superset of the prosodic hierarchies proposed in the literature. The labeling scheme is described in more detail in Price, Ostendorf, Shattuck-Hufnagel, and Fong (1991). Six of the stories were labeled by polling two listeners who discussed any discrepancies. The remaining stories were labeled by a third listener working independently. Comparing the labels of one story using both schemes showed that there was a high degree of consistency across labelers. For the full seven-level labeling system, the correlation between the two sets of labels was 0.93, where correlation is computed as the maximum likelihood estimate of the correlation coefficient based on the two sets of labels. Only 1% of the labels differed by 2, and these were at locations where the disagreement was actually over the location of the boundary rather than the relative strength of the boundary. In this work we considered only a three-level hierarchy and therefore mapped breaks 0–2 to "no break," 3 to a "minor break" (|), 4 and 5 to a "major break" (||) and 6 to a "sentence break."

4.2 Evaluation Methods

The goal of this algorithm is to predict placement of phrase breaks that sound natural to listeners and that communicate the intended meaning of the sentence. As mentioned above, many renditions of a sentence can fulfill this criterion. Therefore, we have attempted to estimate system performance by comparing the predicted breaks to parses observed in five spoken versions of the sentence. Although the ultimate test of the algorithm is in a speech synthesis system, a quantitative measure of system performance is useful in algorithm development and comparison. We have considered four performance measures in this work.

Since one incorrectly assigned break could make a whole sentence or clause unacceptable, one measure of system performance is the number of sentences with a predicted parse that matches entirely a parse observed in any of the five spoken versions. When such a match occurs, we call the predicted parse "correct." The five spoken versions do not represent an exhaustive set of acceptable parses, however. Therefore in a separate evaluation, the sentence is also judged subjectively to determine whether it is an "acceptable" parse. The number of sentences that fall into these two categories are reported separately, and for the best case system are marked separately in the results in the appendix.

In order to better understand the system performance, we have chosen to compute additional error measures based on the prediction accuracy at individual break locations. A predicted sentence is compared to each of the five spoken versions, and the closest spoken version is used as the reference for that sentence. (The closeness of parses is measured using a Euclidean distance with 0 for no break, 1 for minor break and 2 for major break.) Then the correspondence between predicted and observed breaks is tabulated in a confusion matrix. Sentence breaks are deterministically assigned at periods, but these are included in the performance results reported here (as major breaks) to be consistent with results reported elsewhere. Also, note that confusion tables for different systems sometimes reflect different numbers of observed minor and major breaks because the predicted sentences may best match different versions of the test sentence.

It is also useful to have a simple measure for comparing systems. One possible performance figure is the overall percent correct, but we have found this measure to be difficult to interpret because the overall figure is dominated by the performance on the much more frequent "no break" locations. Instead, we compute the correct prediction and false prediction rate for breaks as a combined class (merging minor and major breaks). Using terminology from detection theory, these are also referred to as correct detection (CD) and false detection (FD) in the following sections. CD/FD results must be interpreted with some caution, because there is a trade-off between the two error rates: higher break detection rates are associated with a higher rate of false break insertion. If the insertion rate is too high, there will be few good parses at the sentence level. We have therefore tried to control the insertion rate as much as possible for the different systems evaluated. Two types of CD/FD results are reported. One figure is computed based on comparison to the nearest sentence of the five versions. In addition, since other research results have been reported based on comparison to only one spoken version, we include correct prediction and false prediction rates that correspond to the average rates over the five separate test versions. In general, the correct prediction rates using the single version comparison are roughly 10% lower than using the comparison to five versions, so comparison to one version significantly underestimates performance of the algorithms. The variation in error rate over the five versions is relatively small, as shown later in the discussion of speaker-dependent effects.

4.3 Tree Questions and Designs

Several experiments using different sets of questions to train the embedded decision trees were performed in order to compare the relative merits of different information in the hierarchical model, as well as trade-offs associated with computational complexity. The entire set of questions is listed below. All experiments included questions 1–8, which were based on features that were relatively straightforward to extract from text, using a table look-up to assign part-of-speech labels. Experiments that made use of syntactic features also allowed questions 9–13. The syntax experiments were based on trees that were trained using only 14 of the 20 stories, since skeletal parses were only available for these stories. Another set of experiments included question 14, which tested the ratio of the current minor phrase length to the previous minor phrase length. Finally, experiments that made use of the more detailed POS classifications included question 15, and used the additional particle category in question 3. All questions were based on features derived from text information only.

Below we enumerate the questions used in the different tree design experiments, together with the motivation for each question.

1. *Is this a sentence or major phrase boundary?* Assuming major breaks occur at qualitatively different locations than minor breaks, we effectively remove the major breaks and sentences from our training corpus with this question.
2. *Is the left word a content word and the right word a function word?* In the training data, 65% of the minor and major breaks combined occur at content word/function word (CW/FW) boundaries, and about half of the CW/FW boundaries are marked with breaks. The CW/FW boundaries also correspond to the prosodic group boundaries used deterministically in Sorin, Larreur, and Llorca (1987) and in Veilleux et al. (1990).
3. *What is the function word type of the word to the right?* Previous work in prosodic parsing with a small dictionary (Sorin, Larreur, and Llorca 1987) suggested that different types of function words may be more or less likely to signal a prosodic phrase break.
4. *Is either adjacent word a proper name (capitalized)?* Preliminary examination of our data suggested there was some relationship between proper nouns and phrase boundaries, probably related to the phrasing of complex nominals.
5. *How many content words have occurred since the previous function word?* Speakers seemed to insert phrase breaks when a string of content words became long, e.g., exceeded four or five words.
6. *Is there a comma at this location?* Usually, but not always, a major phrase break occurs at locations orthographically transcribed with commas.
7. *What is the relative location in the sentence (in eighths)?* Previous work (Gee and Grosjean 1983) has suggested that prosodic phrase boundaries tend to bisect a longer unit. Therefore, one of the questions used to partition the training data is the ratio of the word number over the sentence length, quantized to the nearest eighth.
8. *What is the relative location in the proposed major phrase (in eighths)?* This question is included following the same reasoning as the previous question.
9. *What is the largest syntactic unit that dominates the word preceding the potential boundary location and not dominating the succeeding word?* Phrase breaks are known to co-occur with certain syntactic configurations. For example, phrase breaks often occur before subordinate clauses.
10. *What is the largest syntactic unit that dominates the word succeeding the potential boundary location and not dominating the preceding word?* The rationale behind this question is similar to that of the previous question.
11. *What is the smallest syntactic unit that dominates both?* Some syntactic units may be less likely to be broken up by a phrase break.
12. *How many syntactic units end between the two words?* This question provides information on the relative level of syntactic attachment between the two words, capturing the effect of constituent endings.

13. *How many syntactic units begin between the two words?* This question is similar to the previous one, except that it captures effects associated with the start of new constituents.

14. *How large is the ratio of the current minor phrase length over the previous minor phrase length?* This question incorporates the concept of balancing minor phrase lengths noted by other researchers (Gee and Grosjean 1983; Bachenko and Fitzpatrick 1990), and was found to be useful in phrase prediction trees investigated by Wang and Hirschberg (1992). In the beginning of a sentence where there is no previous minor phrase, the ratio is treated as missing data and handled using a surrogate variable (Breiman, Friedman, Olshen, and Stone 1984).

15. *What is the label of the content word to the right? to the left?* Wang and Hirschberg found that part-of-speech information is useful in phrase break prediction (Wang and Hirschberg 1992).

Questions 5, 12, 13, and 14 are based on numerical features, so the binary question asks whether the feature is greater than some threshold, where the threshold is determined automatically in tree design. All other questions are based on categorical variables, and the best binary groupings of the possible values are determined automatically (Breiman, Friedman, Olshen, and Stone 1984). Two of the questions (8, 14) require knowledge of major or minor phrase boundaries. This information is available in the training data or from a spoken utterance, but hypothesized locations of minor and major phrase breaks must be used in phrase prediction from text. Therefore, these features are calculated dynamically in the prediction algorithm for each hypothesized prosodic parse.

The first tree designed used only the very simple information represented by questions 1–8. The resulting tree is shown in Figure 2, with the relative frequency of a break in the training data included at each node. The first split trivially locates the sentence and major break boundaries. The second split utilized the content word/function word boundary question that we had used deterministically in previous work (Veilleux et al. 1990). The content/function word boundaries seem to be important in other algorithms as well: they correspond closely to the phi-phrase boundaries that would be predicted by the Bachenko–Fitzpatrick algorithm, and they seem to be captured in the Wang–Hirschberg text-only tree by a succession of questions about the part-of-speech labels of the words adjacent to the break. Of the boundaries that were preceded by a content word and followed by a function word, 30% were hand-labeled as minor breaks, whereas only 4% of other locations were labeled as minor breaks and these were identified by the next question as coinciding with a comma. The complete tree was relatively small (9 nodes), and used almost all questions provided. On the training data, the resulting tree classified 89% of the nonbreaks correctly and 59% of the minor breaks correctly. All sentence and major breaks were given in the tree design.

The next stage was to incorporate syntactic information (questions 9–13) into the tree design algorithm to determine minor phrase probabilities. Syntactic parses were available for only 14 of the 20 training stories (217 sentences, 4,230 words), and the tree was designed using this subset. A very simple five-node tree was designed, as shown in Figure 3. Again the first nontrivial question was concerning the content/function word boundaries, and the presence of a comma was again used to predict minor breaks at other locations. The two other questions in the tree were based on which syntactic unit dominated one or the other words at the boundary site. The tree design algorithm chose syntactic units that were less likely to contain a boundary as: words

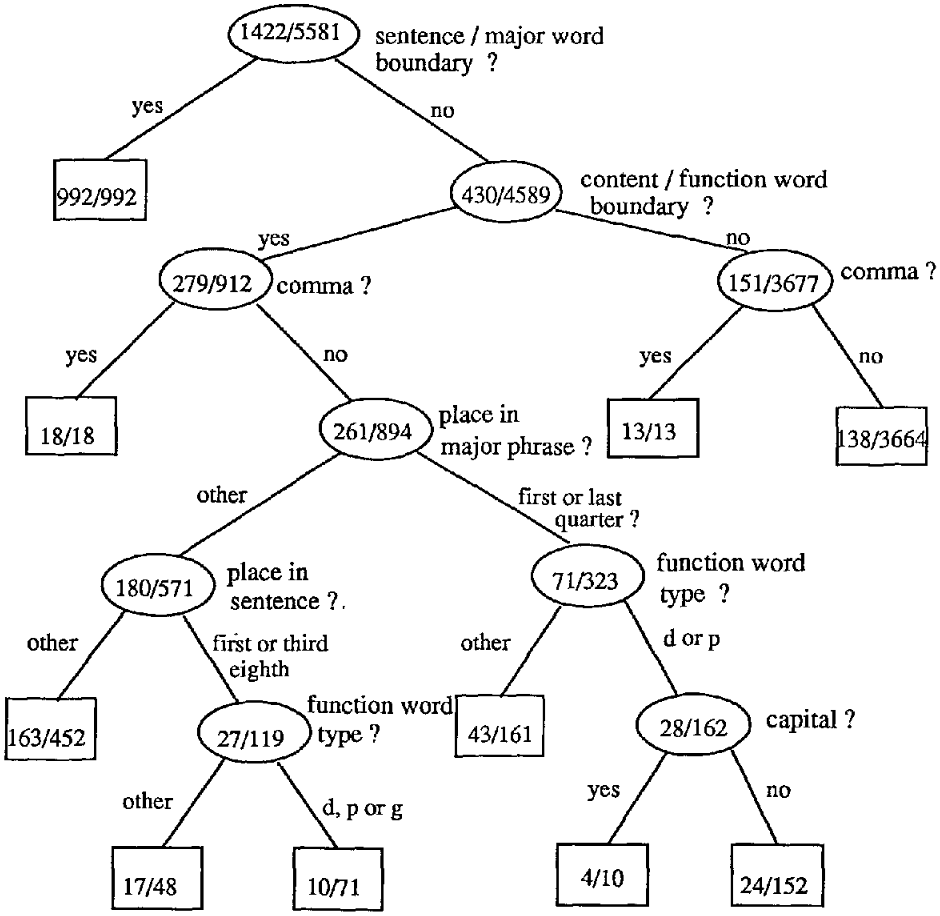

Figure 2
Tree designed using only simple part-of-speech information, questions 1–8. Relative frequency of a "break" (in the training data) is indicated in each node for the subset of data associated with that node, and the left branch in a split is more likely to have a break.

in the same constituent, words separated by a *wh*-noun phrase boundary, and words separated by a verb phrase initial boundary. (In their work on spontaneous speech, Wang and Hirschberg found that noun phrases in general tended to be less likely to contain boundaries.) The tree with syntactic information seemed to classify minor breaks with slightly higher accuracy than the previous tree: 90% correct classification of nonbreaks and 62% correct classification of minor breaks in the training data.

A third tree, illustrated in Figure 4, was grown using the first 8 baseline questions and question 14, which examines the ratio of current minor phrase length to previous minor phrase length. The motivation behind this question is constituent balancing, as mentioned earlier. The main difference between this tree and the first one is that the minor phrase length ratio test is chosen instead of the question about the position in a major phrase. These two questions served similar roles, as evidenced by the fact that the surrogate variable for the ratio test was the location of the current word within

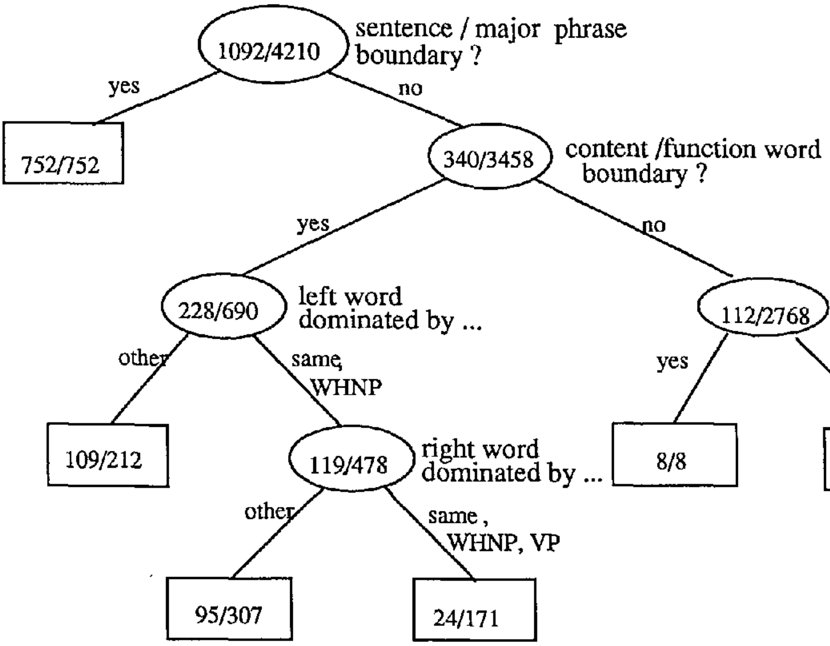

Figure 3
Tree designed using syntactic information but not the minor phrase length ratio test, questions 1–13. Relative frequency of a "break" (in the training data) is indicated in each node for the subset of data associated with that node, and the left branch in a split is more likely to have a break.

the major phrase, in terms of the ratio of the number of words up to the current position over the total length of the major phrase. Classification rates on the training data for this tree were 87% for nonbreaks and 66% for minor breaks. A fourth tree was designed using the first fourteen questions, and performance was similar to that for the third tree.

The decision tree design algorithm's performance was not significantly changed by the introduction of additional features. New features can supplant previously used ones, as also found by Wang and Hirschberg (1992), because of the redundancy in information between features. For example, in the syntax trees, many of the baseline questions were no longer chosen, but the overall classification performance was similar.

4.4 Phrase Prediction Results

The trees were used in the hierarchical model, and the phrase break prediction algorithm was evaluated on the independent test set described in Section 4.1. A summary of the results is given in Table 2, and the corresponding confusion matrices are in Tables 3 and 4. The baseline system (questions 1–8) gave the best performance, with a correct prediction rate of 81% and a false prediction rate of 4%. The results indicate that syntactic information did not improve the performance of the algorithm, and in fact gave poorer phrase predictions by every measure of performance on the test data. The difference in performance cannot be attributed to the smaller amount of training data used in the experiments with syntax, because designing the model without syntax on this subset actually yielded slightly better performance on the test set than that

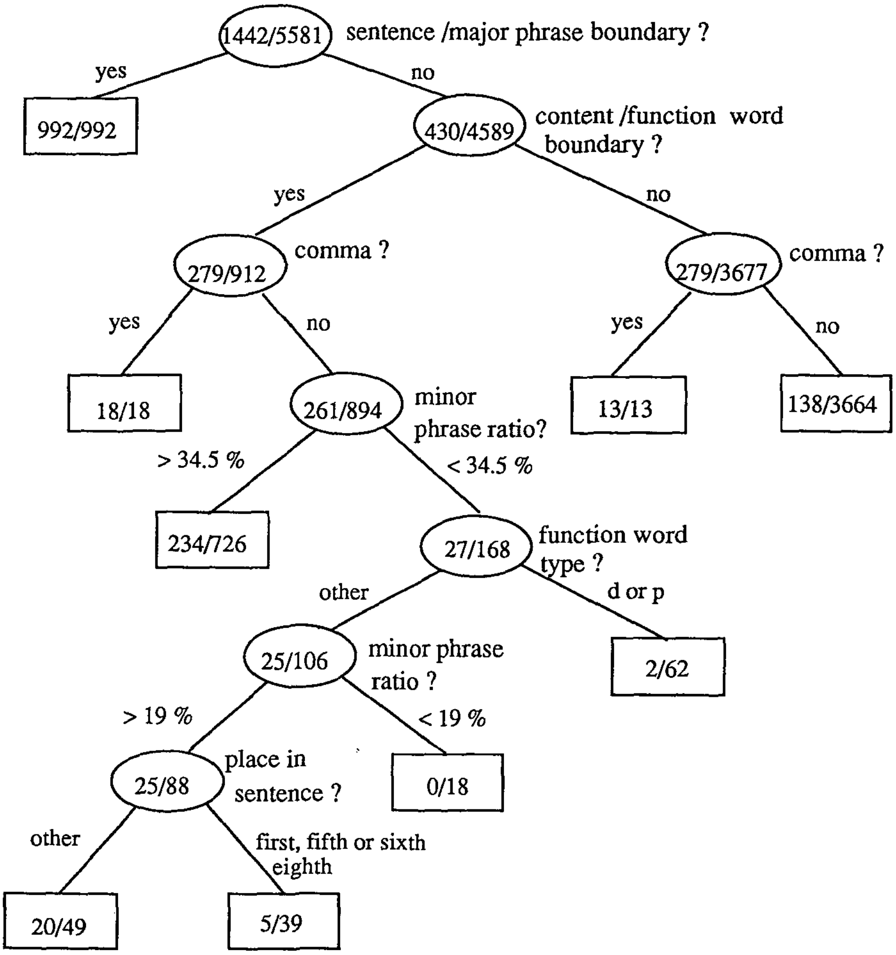

Figure 4
Tree designed using the minor phrase length ratio test but no syntax questions, questions 1–8 and 14. Relative frequency of a "break" (in the training data) is indicated in each node for the subset of data associated with that node, and the left branch in a split is more likely to have a break.

designed on the full training set. We conjecture that the poorer performance associated with using syntax in our model may be due to the fact that syntax plays more of a role in location of major breaks as opposed to the minor breaks predicted in the tree. As we shall see later, syntactic cues were useful in our implementations of the Bachenko–Fitzpatrick and Wang–Hirschberg algorithms. We also found that the minor phrase length ratio test hurt performance, which is likely due to the fact that the ratios are based on hypothesized boundary locations in phrase prediction, as opposed to the known locations used in training. In examining the confusion matrices, we see the main effect of the additional syntactic and minor phrase length ratio questions is more errors at minor phrase boundary locations.

Examining the sentence level performance of the algorithms, we find that a phrase break was inserted between the verb and the particle in three of the six unacceptable

Table 2
Performance of different break prediction algorithms, including variations of our hierarchical model, variations of a tree-based classifier, and the Bachenko–Fitzpatrick (B–F) algorithm based on a test set of 23 sentences (386 words). "Questions Used" refers to those questions listed in Section 4.3 for the two tree-based algorithms. Although the B–F algorithm does not use these specific questions, it does utilize syntactic information as well as relative constituent length. Correct Detection/False Detection (CD/FD) rates are for the merged category of minor and major breaks computing (a) error according to the closest utterance of five versions, and (b) the average error in comparing to a single utterance.

Phrase model	Questions used		Sentences		CD/FD of merged breaks	
	Syntax	Minor ratio	Correct	Accept.	5 versions	1 version
Hierarchical model	No	No	7	11	.81/.04	.70/.05
	Yes	No	4	6	.77/.04	.68/.06
	No	Yes	5	10	.79/.03	.67/.05
	Yes	Yes	3	11	.77/.04	.70/.05
Classification trees	No	No	7	6	.68/.03	.62/.03
	Yes	No	7	5	.72/.01	.61/.01
B–F	Yes	Yes	6	10	.88/.07	.84/.09

Table 3
Confusion matrices for predicted breaks using the hierarchical system with and without syntax, and without the minor phrase length ratio test.

No Syntax			
	Actual		
Predicted	major	minor	no-break
major	49	6	5
minor	5	12	6
no-break	7	10	286

Syntax			
	Actual		
Predicted	major	minor	no-break
major	46	14	11
minor	7	6	1
no-break	12	10	279

Table 4
Confusion matrices for predicted breaks using the hierarchical system with and without syntax, in both cases with the minor phrase length ratio test.

No Syntax			
	Actual		
Predicted	major	minor	no-break
major	51	15	9
minor	5	1	1
no-break	9	9	286

Syntax			
	Actual		
Predicted	major	minor	no-break
major	49	13	9
minor	5	5	2
no-break	9	12	282

parses (e.g., *tried* | *out*, *plugs* | *in*, and *check* | *in*). This is not surprising since the simple POS labeling scheme labels the particle as a preposition. The trees using syntactic information were not able to overcome this effect because of the relative sparsity of particles in the training data (only 5% of the words labeled as prepositions are particles). Other mistakes included a misplaced minor phrase and a deleted major phrase where a comma occurs in the original text. Most of the sentences that were correct (had an exact match with one of the spoken versions) were shorter in length. However, there were several long sentences judged to have acceptable parses. Since many more variations in prosodic phrasing are allowable for longer sentences, it is not surprising that the predicted version was not one of the five spoken versions. The

Table 5
Confusion matrices for predicted breaks using the simple classification trees (no hierarchical model) with and without syntax, in both cases without the minor phrase length ratio test.

No Syntax			
	Actual		
Predicted	major	minor	no-break
major	57	4	8
minor	0	0	0
no-break	13	15	288

Syntax			
	Actual		
Predicted	major	minor	no-break
major	57	3	4
minor	0	0	0
no-break	5	18	298

predicted breaks for the best system, the hierarchical model based on questions 1–8, are shown in the Appendix together with the closest spoken prosodic parse.

In tree design, we chose to represent major breaks as a separate category that the tree was not explicitly designed to detect. A consequence of this choice is that there are fewer "break" data points for training the tree, since there are less than half as many minor breaks as major breaks in the training data. This choice is reasonable if the two breaks occur at qualitatively different locations, which we suspect. In fact, results using trees that were trained by merging major and minor breaks into a single category and then embedded in the hierarchical model had either lower prediction accuracy or a higher false prediction rate. Another consequence of using only minor breaks to train the tree is that features that are associated with major breaks are not represented in the model, which may explain the poor performance of the model with syntax. However, this problem could be addressed with an extension of the current model.

In order to see if explicit modeling of a prosodic hierarchy was a useful aspect of the model, we conducted similar experiments using trees designed specifically for classification. A binary decision tree was trained using the baseline questions (1–8), and another tree was trained using syntax as well (1–13). In both cases, the trees were trained to predict three classes: major break, minor break, and no break. Including the minor phrase length ratio test using known break locations (from hand labels) did not improve prediction performance, so we did not implement a dynamic version based on hypothesized minor breaks. Results for these two trees are included in Table 2, with confusion matrices given in Table 5. The costs of different errors were chosen to obtain a false detection rate similar to that for the hierarchical model. Choosing good costs proved difficult, so the correct detection rate is lower than that for the other models primarily because the false detection rate was so low. The difference in false detection rates makes comparison to the hierarchical model difficult. However, experience with performance of the models at different false detection rates suggests that the baseline hierarchical model outperforms the classification tree that does not use syntax, but that the classification tree that uses syntax is at least as good as the hierarchical model. Since the complexity associated with obtaining a syntactic parse is significantly greater than that associated with the simple three-level hierarchy that we have proposed, we conclude that explicit modeling of a hierarchy is a useful feature of the model. In addition, the fact that syntactic information was useful for the classification tree but not for the hierarchical model suggests that syntactic features are more important for predicting major breaks than minor breaks, since major breaks are not represented as a class in tree design for the hierarchical model.

For both the hierarchical model and the simple classification trees, we also investigated the use of more detailed part-of-speech information both with and without

Table 6
Confusion matrix for the Bachenko–Fitzpatrick algorithm, which uses syntax and rules to account for balancing constituent lengths.

	Actual		
Predicted	major	minor	no-break
major	49	9	7
minor	10	21	12
no-break	4	8	265

the syntactic features. The more detailed part-of-speech did not improve performance under any of these conditions. For the classification trees, correct detection improved slightly, but there was a corresponding increase in false detection. For the hierarchical model, performance actually degraded. These results suggest that the complexity associated with more detailed part-of-speech tagging may not be necessary; however, further research is needed to answer this question. It may be that other POS questions, such as testing a larger window of words around the break as in Wang and Hirschberg (1992), would yield better results.

Finally, we thought it would be interesting to compare the results of our prediction algorithm to that of Bachenko and Fitzpatrick's algorithm on our corpus. Since the test set was relatively small, we were able to implement the algorithm by predicting the phrase boundaries from the rules by hand. To assign node indices to prosodic breaks (Bachenko and Fitzpatrick 1990), a critical value for separating major and minor phrase breaks is calculated based on an average of the indices associated with the prosodic phrase nodes, where the prosodic phrase nodes are all those created by the Bachenko–Fitzpatrick primary salience rules. Boundaries with an index greater than the critical value are assigned a major break, indices below 5 have no prosodic break, and intermediate indices map to a minor prosodic break. (Bachenko and Fitzpatrick include index 4 in the minor break category, but 5 was used here to obtain a lower insertion rate.) For multiple verb phrases in sequence, the verb balancing rule is applied left-to-right until all verb phrases are grouped before applying the verb adjacency rule or other processing. The confusion matrix for these results is shown in Table 6, and the performance summary is also included in Table 2. Although the correct break detection rate is significantly higher than that for the other algorithms, the false detection rate is also higher, and so the sentence accuracy is similar to that for the baseline hierarchical model. Unlike the other algorithms, the Bachenko–Fitzpatrick algorithm did not make the mistake of assigning a minor phrase break before a particle, but this relies on having a parser that can make that distinction. An advantage of both the classification tree and the hierarchical model over the Bachenko–Fitzpatrick model is that they can be automatically trained, and thus can be tuned to handle particular tasks.

Table 7 gives the correct detection and false detection rates calculated by comparing the predicted prosodic parses to each of the different spoken versions. The performance of speaker f2b, whose speech made up roughly three-quarters of the training data, had performance similar to the average for the five versions with slightly lower correct detection rates but also slightly lower false detection rates. These results suggest that the automatic algorithms are not particularly speaker-dependent, though we expect that it is important to have similar styles for both training and test data. There was no consistent difference in performance between male and female speakers, and the difference in error rates for different speakers was relatively small.

Table 7
Correct detection/false detection rates for predicted phrase breaks to each of the five different test versions. With the exception of the Bachenko–Fitzpatrick algorithm, the systems included here did not use the minor ratio question. Speaker codes begin with "f" or "m" for female and male speakers, respectively. Speaker code f2b is annotated with "r" for the original radio recording and "l" for the subsequent lab recording.

Phrase model	Uses syntax?	CD/FD of merged breaks					
		f2b(r)	f2b(l)	f3a	m1b	m3b	average
Hierarchical model	No	.69/.04	.66/.05	.73/.06	.72/.06	.70/.06	.70/.05
	Yes	.66/.06	.69/.05	.68/.08	.71/.07	.69/.07	.68/.06
Classification trees	No	.60/.02	.59/.02	.64/.04	.65/.04	.62/.04	.62/.03
	Yes	.58/.01	.57/.01	.61/.02	.67/.01	.63/.01	.61/.01
B-F	Yes	.85/.06	.82/.07	.81/.11	.85/.10	.85/.10	.84/.09

Table 8
Comparison of correct detection/false detection rates computed from five test versions for different break prediction algorithms that do and do not use comma information.

	Hierarchical Model		Classification Trees	
	baseline	all features	baseline	w/syntax
Commas	.81/.04	.77/.04	.68/.03	.72/.01
No commas	.66/.05	.71/.04	.59/.04	.68/.02

In all of the previous experiments, the presence of a comma was an important feature for predicting phrase boundaries for all algorithms implemented. While this is a valid feature in text-to-speech synthesis applications, it is not available in applications involving spoken speech. (Although presence of a pause might be a useful alternative feature.) In addition, as we have mentioned earlier, commas are not reliably used even in written text. Therefore, it is interesting to determine the performance of the algorithm without the comma feature. As expected, performance degrades significantly, both for the hierarchical model and for the classification trees. In addition, for the hierarchical model, syntactic information and the minor phrase length ratio test now provide information that improves performance over the baseline system. To illustrate performance differences, some correct prediction/false prediction rates are given in Table 8.

It is difficult to compare our performance figures with other reported results because of differences in corpora and speaking styles. However, the average single speaker correct detection and false detection rates reported here for our implementations of the Bachenko–Fitzpatrick and Wang–Hirschberg algorithms indicate the robustness of these algorithms to different types of data. Our results for the Bachenko–Fitzpatrick algorithm are somewhat higher than those that they report, .84/.09 vs. .78/.08.[2] Using only the features inferable from text, Wang and Hirschberg use classification trees to predict prosodic boundaries in spontaneous speech, achieving phrase break prediction results of .66/.02.[3] (Again, note that these results are not directly comparable because of the differences in false detection rates, and results for other trees in Wang and Hirschberg [1992] suggest that these two algorithms have similar

[2] This figure is calculated from the examples in the appendix in Bachenko and Fitzpatrick (1990), ignoring tertiary boundaries and including sentence-final boundaries as correct. The sentence that did not parse was not included in the calculation.

[3] This result is computed from Figure 6 of Wang and Hirschberg (1992), which illustrates classification on training data. Cross validation results may vary slightly.

performance.) Our classification trees used somewhat different features, though also based on POS and syntactic information, and achieved results on radio news speech that are surprisingly lower, i.e., .59/.02 for the tree that used syntax but did not use commas. Of course, the POS and syntactic information used here may not have been as detailed and/or reliable as that used by Wang and Hirschberg. (The comparisons here are based on the average error rates for single version comparisons.)

5. Discussion

In summary, the model proposed here addresses several issues in modeling prosodic phrase structure. The model is a general formalism for an embedded hierarchy, which represents a unit in terms of the probability of the sequence of subunits comprising it. The model is specifically applied to represent a hierarchy containing sentences, major phrases, and minor phrases. The model captures grammatical and constituent length factors through the use of a decision tree, as in Wang and Hirschberg (1992), but embedding the tree within a hierarchical structure yields better performance than that achieved by a decision tree alone. The model is stochastic, which accounts for the natural variability in prosodic parsing, and can be automatically trained to reflect different speaking styles. The automatic training algorithm described here, based on maximum likelihood estimation, involves simple relative frequency estimates and decision tree design. Automatic training on a large corpus to design the best predictors of phrase breaks can also provide new insight into the relationship between prosody and syntax. Using the stochastic model, prosodic phrase break prediction involves choosing the most likely prosodic parse, which can be achieved using a recursive dynamic programming algorithm. We have found that good phrase break prediction results can be achieved without the use of syntactic information or detailed part-of-speech labels, resulting in a very low complexity prediction algorithm. Without syntax, our algorithm predicted a good prosodic parse for 18 out of 23 sentences, which corresponds to a correct break prediction of 81% and a false prediction rate of 4%.

There are many ways in which this work could be extended. As we have pointed out, it would be useful to use features directly in determining the probability of a unit, rather than simply representing a unit in terms of the probabilities of the subunits. For example, we conjecture that commas and certain syntactic structures might be good predictors of major phrase breaks, but not minor phrase breaks. In addition, other features could be used in the model, including different syntactic features and different questions about the more detailed part-of-speech labels. Automatically predicted prominence (or pitch accent) locations might also be useful in phrase boundary prediction, although it is arguable whether prominence prediction should come before or after boundary placement. Of course, it would also be interesting to consider a higher order hierarchy, though we anticipate that a successful implementation would require representation of features at the different levels. The results reported here were limited to some extent by the amount of available data. A larger training set would enable the study of more factors, including possibly paragraph-level phenomena. A larger test set would better establish the significance of the results. Finally, the best test of a phrase break prediction algorithm is in perceptual judgments of synthesized speech, and we would like to evaluate our algorithm in this context.

In this work we have focused on the synthesis application of prosodic phrase break prediction. However, one of the advantages of a stochastic model is that it may be useful for analysis of spoken speech. Because there is some relationship between prosody and syntax, prosodic phrase structure can be used to improve the performance and/or speed of speech understanding systems. For example, a score of the consistency between a syntactic parse and a prosodic parse has been used to resolve ambiguities in sentence interpretation, where the score is computed by comparing automatically detected prosodic phrase breaks to phrase breaks predicted from the text of the different interpretations (Ostendorf, Wightman, and Veilleux 1993). Another approach to syntactic disambiguation using prosodic information is described in Bear and Price (1990) and Ostendorf, Price, Bear, and Wightman (1990), where prosodic breaks are used to constrain grammar rules in a parser. A stochastic phrase model could also be used to improve performance of this system simply by serving as a "language model" (as in speech recognition) to improve the performance of a detection algorithm using acoustic information.

Acknowledgments

We would like to thank Patti Price and Stefanie Shattuck-Hufnagel for their useful comments and insights, an anonymous reviewer for many valuable comments, and Gay Baldwin and Liz Shriberg for hand-labeling part of the corpus with prosodic labels. The radio news corpus was provided by WBUR, a public radio station. The skeletal parses for the corpus are part of a preliminary version of the Penn Treebank Corpus, copyright of the University of Pennsylvania. This work was supported by NSF under grant number IRI-8805680.

References

Allen, J.; Hunnicutt, S.; Carlson, R.; and Granstrom, B. (1979). "MITalk-79: The 1979 MIT text-to-speech system." In *Speech Communications Papers Presented at the 97th Meeting of the ASA*, edited by Wolf and Klatt, 507–510.

Allen, J.; Hunnicutt, M. S.; and Klatt, D. (1987). *From Text to Speech: The MITalk System.* Cambridge University Press.

Altenberg, B. (1987). *Prosodic Patterns in Spoken English.* Lund University Press.

Bachenko, J., and Fitzpatrick, E. (1990). "A computational grammar of discourse-neutral prosodic phrasing in English." *Computational Linguistics* 16(3), 155–170.

Bahl, L. R.; Brown, P. F.; deSouza, P. V.; and Mercer, R. L. (1989). "A tree-based statistical language model for natural language speech recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(7), 1001–1008.

Bear, J., and Price, P. J. (1990). "Prosody, syntax and parsing." In *Proceedings, ACL Conference*, 17–22.

Beckman, M., and Pierrehumbert, J. (1986). "Intonational structure in Japanese and English." In *Phonology Yearbook 3*, edited by J. Ohala, 255–309.

Booij, G. (1983). "Principles and parameters in prosodic phonology." *Linguistics* 21, 249–280.

Breiman, L.; Friedman, J.; Olshen R.; and Stone, C. (1984). *Classification and Regression Trees*, The Wadsworth Statistics/Probability Series, Wadsworth and Brooks.

Church, K. W. (1988). "A stochastic parts program and noun phrase parser for unrestricted text." In *Proceedings, Second Conference on Applied Natural Language Processing*, 136–143.

Cooper, W., and Paccia-Cooper, J. (1980). *Syntax and Speech.* Harvard University Press.

Gee, J., and Grosjean, F. (1983). "Performance structures: A psycholinguistic and linguistic appraisal." *Cognitive Psychology* 15, 411–458.

Hindle, D. M. (1987). "Acquiring disambiguation rules from text." In *Proceedings, Association for Computational Linguistics Meeting*, 118–125.

Hirose, K., and Fujisaki, H. (1982). "Analysis and synthesis of voice fundamental frequency contours of spoken sentences." In *Proceedings, International Conference on Acoustics, Speech, and Signal Processing*, 950–953.

Ladd, D. R. (1986). "Intonational phrasing: The case for recursive prosodic structure." In *Phonology Yearbook 3*, edited by J. Ohala, 311–340.

Ladd, D. R., and Campbell, N. (1991). "Theories of prosodic structure: Evidence from syllable duration." In *Proceedings, XII International Congress of Phonetic Sciences*, 2, 290–293.

Lehiste, I. (1973). "Phonetic disambiguation of syntactic ambiguity." *Glossa* 7(2), 107–121.

Liberman, M. Y., and Prince, A. S. (1977). "On stress and linguistic rhythm." *Linguistic Inquiry* 8, 249–336.

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. (1993). "Building a very large annotated corpus of English:

The Penn Treebank." *Computational Linguistics* 19(2), 313–330.

Meteer, M.; Schwartz, R.; and Weischedel, R. (1991). "POST: Using probabilities in language processing." In *Proceedings, International Joint Conference on Artificial Intelligence,* 960–965.

Nespor, M., and Vogel, I. (1983). "Prosodic structure above the word." In *Prosody: Models and Measurements,* edited by A. Cutler and D. R. Ladd, 123–140. Springer-Verlag.

Nespor, M., and Vogel, I. (1986). *Prosodic Phonology.* Foris.

O'Shaughnessy, D. (1989). "Parsing with a small dictionary for applications such as text-to-speech." *Computational Linguistics* 15(2), 97–108.

Ostendorf, M.; Price, P.; Bear, J.; and Wightman, C. W. (1990). "The use of relative duration in syntactic disambiguation." In *Proceedings, Third DARPA Workshop on Speech and Natural Language,* 26–31.

Ostendorf, M., and Roukos, S. (1989). "A stochastic segment model for phoneme-based continuous speech recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(12), 1857–1869.

Ostendorf, M.; Wightman, C. W.; and Veilleux, N. M. (1993). "Parse scoring with prosodic information: An analysis/synthesis approach." *Computer Speech and Language,* 193–210.

Price, P.; Ostendorf, M.; Shattuck-Hufnagel, S.; and Fong, C. (1991). "The use of prosody in syntactic disambiguation." *Journal of the Acoustical Society of America* 90(6), 2956–2970.

Selkirk, E. (1980). "The role of prosodic categories in English word stress." *Linguistic Inquiry* 11, 563–605.

Selkirk, E. (1984). *Phonology and Syntax: The Relation between Sound and Structure.* MIT Press.

Sorin, C.; Larreur, D.; and Llorca, R. (1987). "A rhythm-based prosodic parser for text-to-speech systems in French." In *Proceedings, International Congress of Phonetic Sciences* 1, 125–128.

t'Hart, J.; Collier, R.; and Cohen, A. (1990). *A Perceptual Study of Intonation.* Cambridge University Press.

Veilleux, N.; Ostendorf, M.; Price, P.; and Shattuck-Hufnagel, S. (1990). "Markov modeling of prosodic phrase structure." In *Proceedings, International Conference on Acoustics, Speech, and Signal Processing,* 777–780.

Wang, M., and Hirschberg, J. (1992). "Automatic classification of intonational phrase boundaries." *Computer Speech and Language* 6(2), 175–196.

Wightman, C.; Shattuck-Hufnagel, S.; Ostendorf, M.; and Price, P. (1992). "Segmental durations in the vicinity of prosodic phrase boundaries." *Journal of the Acoustical Society of America* 91(3), 1707–1717.

Appendix: Predicted Breaks in Test Corpus

The sentences below illustrate the predicted minor (|) and major (||) phrase boundaries for our best case algorithm, that which uses the basic questions 1–8 in the hierarchical model. The evaluations "correct" (C), "acceptable" (A), and "incorrect" (I) are indicated after each sentence. For the cases where the predicted sentences were either acceptable or incorrect, we have included the spoken version that was closest to the predicted sentence in the sense of minimizing the Euclidean distance based on representing no break with 0, a minor break with 1, and a major break with 2.

1. Computerized phone calls, || which do everything from selling magazine subscriptions | to reminding people about meetings, || have become the telephone equivalent || of junk mail. || (A)
   Computerized phone calls, || which do everything | from selling magazine subscriptions || to reminding people about meetings || have become the telephone equivalent | of junk mail. ||
2. But a new application of the technology || is about to be tried | out in Massachusetts || to ease crowded jail conditions. || (I)
   But a new application of the technology || is about to be tried out in Massachusetts || to ease crowded jail conditions. ||
3. Next week some inmates released early | from the Hampton County jail in Springfield || will be wearing a wristband that hooks up with a special jack | on their home phones. || (A)
   Next week || some inmates released early | from the Hampton County jail in Springfield || will be wearing a wristband || that hooks up with a special jack | on their home phones. ||
4. Whenever a computer randomly calls || them from jail, || the former prisoner plugs | in to let corrections officials know || they're in the right place | at the right time. || (I)
   Whenever a computer randomly calls them from jail, || the former prisoner plugs in | to let corrections officials know || they're in the right place | at the right time. ||
5. Margo Melnicove reports. || (C)
6. The device is attached | to a plastic wristband. || (C)
7. It looks like a watch. || (C)
8. It functions like an electronic probation officer. || (A)
   It functions | like an electronic probation officer. ||
9. When a computerized call is made | to a former prisoner's home phone, || that person answers | by plugging in the device. || (C)
10. The wristband can be removed || only by breaking its clasp, || and if that's done the inmate | is immediately returned to jail. || (A)
    The wristband | can be removed || only by breaking its clasp, || and if that's done || the inmate | is immediately returned to jail. ||
11. The description conjures up images | of big brother watching. || (C)
12. But Jay Ash, || deputy superintendent of the Hampton County jail | in Springfield, | says the surveillance system || is not that sinister. || (I)

But Jay Ash, || deputy superintendent of the Hampton County jail in Springfield, || says the surveillance system | is not that sinister. ||

13. Such supervision, || according to Ash, | is a sensible, || cost effective alternative to incarceration | that should not alarm civil libertarians. || (A)
Such supervision, || according to Ash, || is a sensible, || cost effective alternative to incarceration || that should not alarm || civil libertarians. ||

14. Doctor Norman Rosenblatt, | dean of the college | of criminal justice at Northeastern University, || agrees. || (A)
Doctor Norman Rosenblatt, || dean of the college | of criminal justice at Northeastern University, || agrees. ||

15. Rosenblatt expects electronic surveillance | in parole situations to become more widespread, || and he thinks eventually people || will get used to the idea. || (I)
Rosenblatt expects electronic surveillance in parole situations to become more widespread, || and he thinks eventually || people will get used to the idea. ||

16. Springfield jail deputy superintendent Ash says || although it will allow || some prisoners to be released || a few months || before their sentences are up, || concerns that may raise about public safety || are not well founded. || (A)
Springfield jail deputy superintendent Ash | says || although it will allow | some prisoners to be released | a few months before their sentences are up, || concerns that may raise | about public safety || are not well founded. ||

17. Most county jail inmates || did not commit violent crimes. || (C)

18. They're in jail for such things | as bad checks or stealing. || (A)
They're in jail for such things | as bad checks | or stealing. ||

19. Those on early release must check | in with corrections officials fifty times || a week according to Ash,|| who says about half | the contacts for a select group || will now be made || by the computerized phone calls. || (I)
Those on early release | must check in with corrections officials || fifty times a week || according to Ash,|| who says about half | the contacts for a select group || will now be made | by the computerized phone calls. ||

20. Initially the program will involve || only a handful of inmates. || (A)
Initially | the program will involve | only a handful of inmates. ||

21. Ash says the ultimate goal || is to use it to get | about forty out of jail early. || (A)
Ash says | the ultimate goal || is to use it to get about forty || out of jail early. ||

22. The Springfield jail, || built for 270 people, | now houses more than 500. || (A)
The Springfield jail, || built for 270 people, || now houses more than 500. ||

23. For WBUR, || I'm Margo Melnicove. || (C)