Some Applications of Tree-based Modelling to Speech and Language

Michael D. Riley

AT&T Bell Laboratories

1. Introduction

Several applications of statistical tree-based modelling are described here to problems in speech and language. Classification and regression trees are well suited to many of the pattern recognition problems encountered in this area since they (1) statistically select the most significant features involved (2) provide "honest" estimates of their performance, (3) permit both categorical and continuous features to be considered, and (4) allow human interpretation and exploration of their result. First the method is summarized, then its application to automatic stop classification, segment duration prediction for synthesis, phoneme-to-phone classification, and end-of-sentence detection in text are described. For other applications to speech and language, see [Lucassen 1984], [Bahl, et al 1987].

2. Classification and Regression Trees

An excellent description of the theory and implementation of tree-based statistical models can be found in *Classification and Regression Trees* [L. Breiman, et al, 1984]. A brief description of these ideas will be provided here.

Figure 1 shows an example of a tree for classifying whether a stop (in the context C V) is voiced or voiceless based on factors such as voice onset time, closure duration, and phonetic context. Let us first see how to use such a tree for classification. Then we will see how the tree was generated.

Suppose we have a stop with a VOT of 30 msec that is preceded by a nasal and followed by a high back vowel. Starting at the root node in Figure 1, the first decision is whether the VOT is greater or less than 35.4 msec. Since in our example, it is less, we take the left branch. The next split, labelled "l-cm", refers to the consonantal manner of the the preceding (left) segment. Since in this case it is nasal, we take the right branch. The next split is on the vowel place of following (right) segment. Since it is high back in this case, we take the right branch, reaching a terminal node. The node is labelled "yes", indicating that this example should be classified as voiced.

In the training set, 739 of the 1189 examples that reached this node were correctly classified. This tree is a subtree of a better classifier to be described in the next section; this example was pruned for illustrative purposes.

This is an example of a *classification* tree, since the decision is to choose one of several classes; in this case, there are two classes: $\{voiced, voiceless\}$. In other words, the predicted variable, y, is categorical. Trees can be created for continuous y also. In this case they are called regression trees with the terminal nodes labelled with a real number (or, more generally, a vector).

Classifying with an existing tree is easy; the interesting question is how to generate the tree for a given problem. There are three basic questions that have to be answered when generating a tree: (1) what are the splitting rules, (2) what are the stopping rules, and (3) what prediction is made at each terminal node?

Voiced or Voiceless Stop?

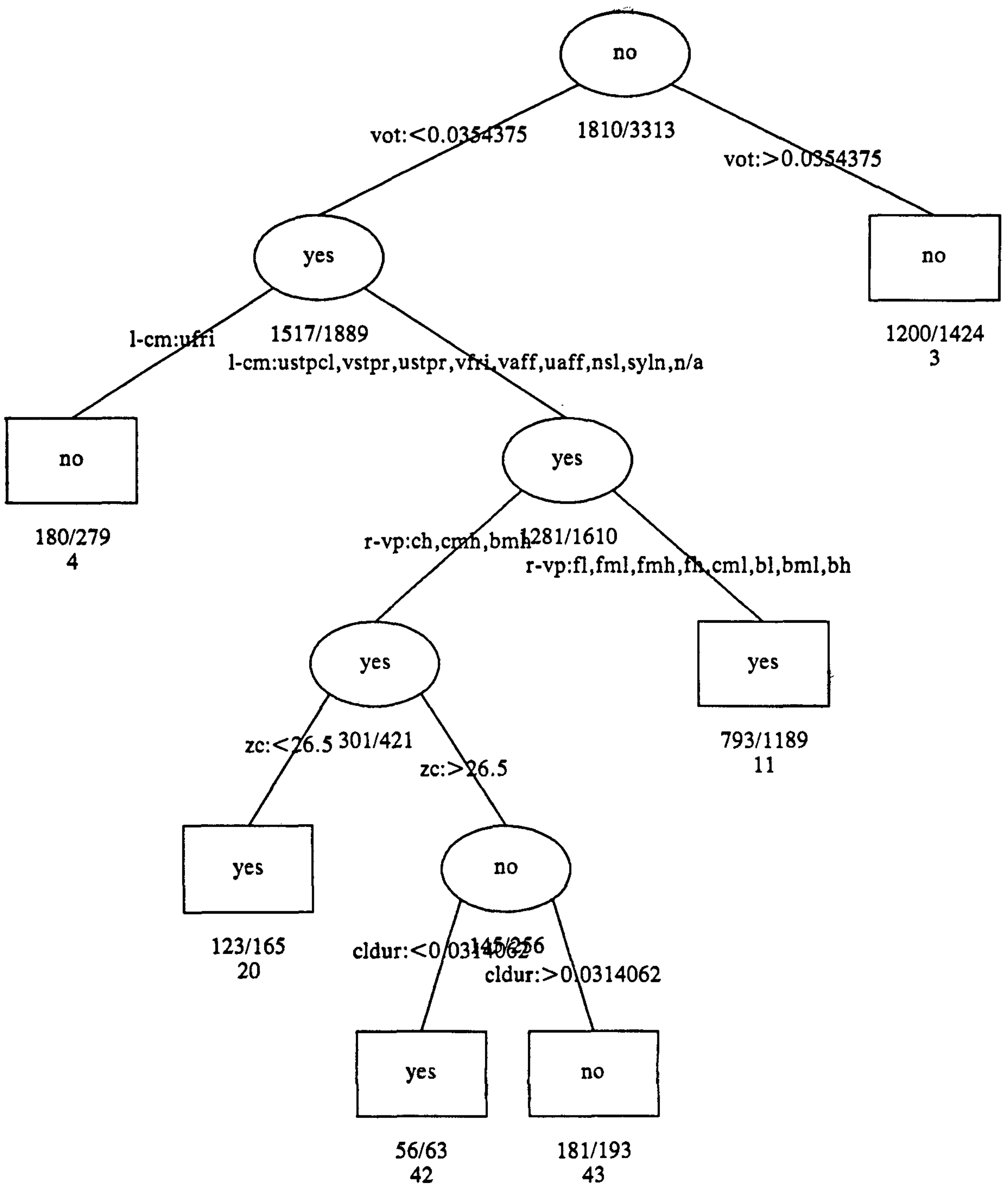


Figure 1. *Classification tree for voiced vs voiceless stop.*

Let us begin answering these questions by introducing some notation. Consider that we have N samples of data, with each sample consisting of M features, $x_1, x_2, x_3, \ldots x_m$. In the voiced/voiceless stop example, x_1 might be VOT, x_2 the phonetic class of the preceding segment, etc. Just as the y (dependent) variable can be continuous or categorical, so can the x (independent) variables. E.g., VOT is continuous, while phonetic class is categorical (can not be usefully ordered).

The first question — what stopping rule? — refers to what split to take at a given node. It has two parts: (a) what candidates should be considered, and (b) which is the best choice among candidates for a given node?

A simple choice is to consider splits based on one x variable at a time. If the independent variable being considered is continuous $-\infty \leq x < \infty$, consider splits of the form:

$$x \leq k \quad \text{vs.} \quad x > k, \quad \forall k.$$

In other words, consider all binary cuts of that variable. If the independent variable is categorical $x \in \{1, 2, ..., n\} = X$, consider splits of form:

$$x \in A \quad \text{vs.} \quad x \in X - A, \quad \forall A \subset X.$$

In other words, consider all binary partitions of that variable. More sophisticated splitting rules would allow combinations of a such splits at a given node; e.g., linear combinations of continuous variables, or boolean combinations of categorical variables.

A simple choice to decide which of these splits is the best at a given node is to select the one that minimizes the estimated classification or prediction error after that split based on the training set. Since this is done stepwise at each node, this is not guaranteed to be globally optimal even for the training set.

In fact, there are cases where this is a bad choice. Consider Figure 2, where two different splits are illustrated for a classification problem having two classes (No. 1 and No. 2) and 800 samples in the training set (with 400 in each class). If we label each child node according to the greater class present there, we see that the two different splits illustrated both give 200 samples misclassified. Thus, minimizing the error gives no preference to either of these splits.

The example on the right, however, is better because it creates at least one very pure node (no misclassification) which needs no more splitting. At the next split, the other node can be attacked. In other words, the stepwise optimization makes creating purer nodes at each step desirable. A simple way to do this is to minimize the entropy at each node for categorical y. Minimizing the mean square error is a common choice for continuous y.

The second question — what stopping rule? — refers when to declare a node terminal. Too large trees may match the training data well, but they won't necessarily perform well on new test data, since they have overfit the data. Thus, a procedure is needed to find an "honest-sized" tree.

Early attempts at this tried to find good stopping rules based on absolute purity, differential purity from the parent, and other such "local" evaluations. Unfortunately, good thresholds for these vary from problem to problem.

A better choice is as follows: (a) grow an over-large tree with very conservative stopping rules, (b) form a sequence of subtrees, $T_0, \ldots, T_n$, ranging from the full tree to just the root node, (c) estimate an "honest" error rate for each subtree, and then (d) choose the subtree with the minimum "honest" error rate.

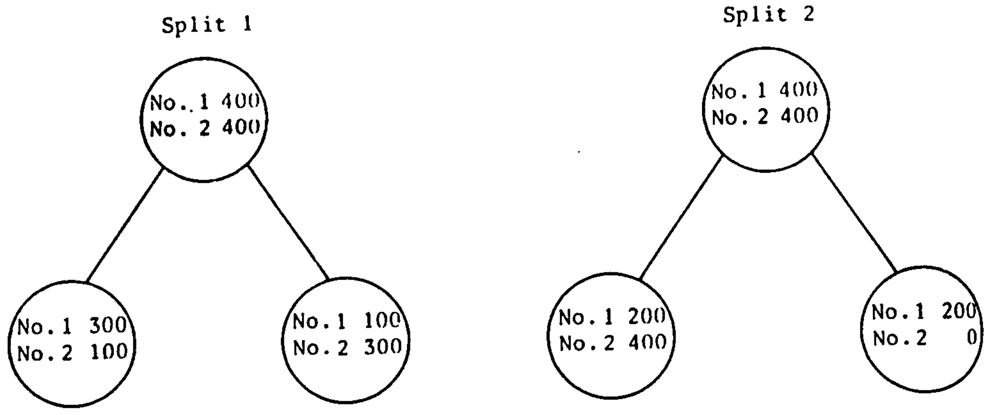


Figure 2. *Two different splits with the same misclassification rate.*

To form the sequence of subtrees in (b), vary α from 0 (for full tree) to ∞ (for just the root node) in:

$$\min_T [R(T) + \alpha |T|].$$

where $R(T)$ is the classification or prediction error for that subtree and $|T|$ is the number of terminal nodes in the subtree. This is called the cost-complexity pruning sequence.

To estimate an "honest" error rate in (c), test the subtrees on data different from the training data, e.g., grow the tree on 9/10 of the available data and test on 1/10 of the data repeating 10 times and averaging. This is often called cross-validation.

Figure 3 shows misclassification rate vs. tree length for the voiced-voiceless stop classification problem. The bottom curve shows misclassification for the training data, which continues to improve with increasing tree length. The higher curve shows the cross-validated misclassification rate, which reaches a minimum with a tree size of about 30 and then rises again with increasing tree length. In fact a tree length of around 10 is very near optimal and would be a good choice for this problem.

The last question — what prediction is made at a terminal node? — is easy to answer. If the predicted variable is categorical, choose the most frequent class among the training samples at that node (plurality vote). If it is continuous, choose the mean of the training samples at that node.

The approach described here can be used on quite large problem. We have grown trees with hundreds of thousands of samples with a hundred different independent variables. The time complexity, in fact, grows only linearly with the number of input variables. The one expensive operation is forming the binary partitions for categorical $x's$. This increases exponentially with the number of distinct values the variable can assume.

Let us now discuss some applications of these ideas to some problems in speech and language.

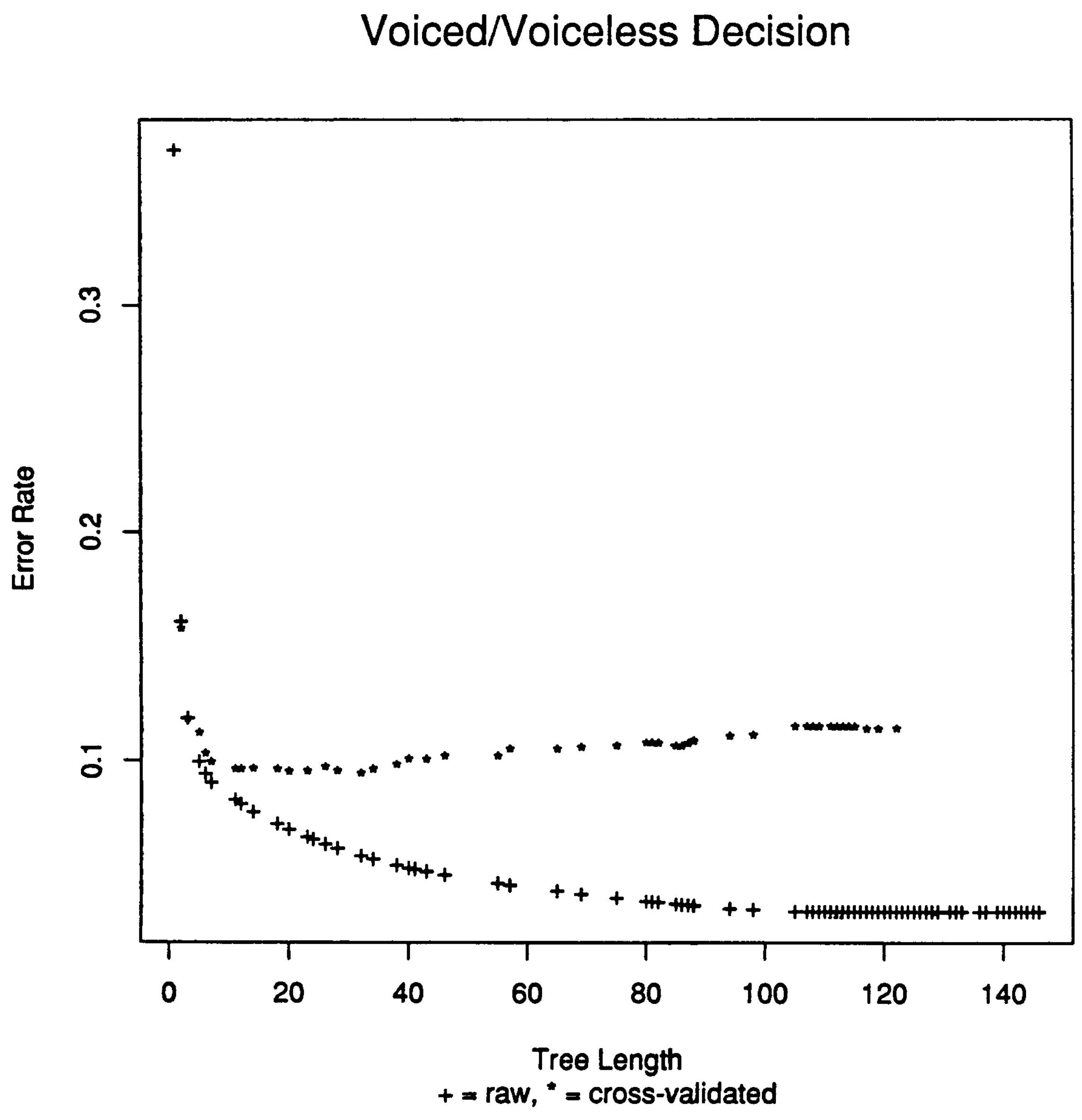

Voiced/Voiceless Decision
Error Rate
0.1
0.2
0.3
0
20
40
60
80
100
120
140
Tree Length
+ = raw, * = cross-validated

Figure 3.

3. Stop Classification

The first application has already partially been introduced. Figure 4 shows a more complete tree than Figure 1 for deciding whether a stop is voiced or voiceless. This tree size was selected for the reasons given above. This tree was grown from 3313 *stop+vowel* examples taken from male speaker in the TIMIT database. The classification task is to decide whether a given stop was labelled voiced *(b, d, g)* or unvoiced *(p, t, k)* by the TIMIT transcribers.

The features (possible $x's$) considered were:

- Voice Onset Time
- Closure Duration
- Vowel Duration
- Zero Crossing Rate between Release and Onset
- Phonetic Context:
  - Segment to left: manner/place, consonant/vowel
  - Segment to right: manner/place
- Vowel Formant Freqs. and Slopes at Onset
- F0 and F0 Slope at Onset

The first three features were computed directly from the TIMIT labellings. The zero-crossing rate was the mean rate between release and onset. The formant and F0 values were computed using David Talkin's formant and pitch extraction programs [Talkin 1987].

Coding the phonetic context required special considerations since more than 50 phones (using the TIMIT labelling) can precede a stop in this context. If this were treated as a single feature, more than 2^{50} binary partitions would have to be considered for this variable at each node, clearly making this approach impractical. Chou [1987] proposes one solution, which is to use k-means clustering to find sub-optimal, but good paritions in linear complexity.

The solution adopted here is to classify each phone in terms of 4 features, *consonant manner, consonant place, "vowel manner", and "vowel place"*, each class taking on about a dozen values. Consonant manner takes on the usual values such as *voiced fricative, unvoiced stop, nasal, etc.* Consonant manner takes on values such as *bilabial, dental, velar, etc.* "Vowel manner" takes on values such as *monopthong, diphthong, glide, liquid, etc.* and "vowel place" takes on values such as *front-low, central-mid-high, back-high, etc.* All can take on the value "n/a" if they do not apply; e.g., when a vowel is being represented, consonant manner and place are assigned "n/a". In this way, every segment is decomposed into four multi-valued features that have acceptable complexity to the classification scheme and that have some phonetic justification.

The tree in Figure 4 correctly classifies about 91% of the stops as voiced or voiceless. All percent figures quoted in this paper are cross-validated unless otherwise indicated. In other words, they are tested on data distinct from the training data.

In an informal experiment, the author listened to 1000 of these stops cut at 30 msec before the stop closure and at 30 msec after the vowel onset. He correctly classifed 90% of these stops as voiced or voiceless. This suggests that the input features selected were appropriate to the task and that the classification tree was a reasonable structure to exploit the information carried by them.

Voiced or Voiceless Stop?

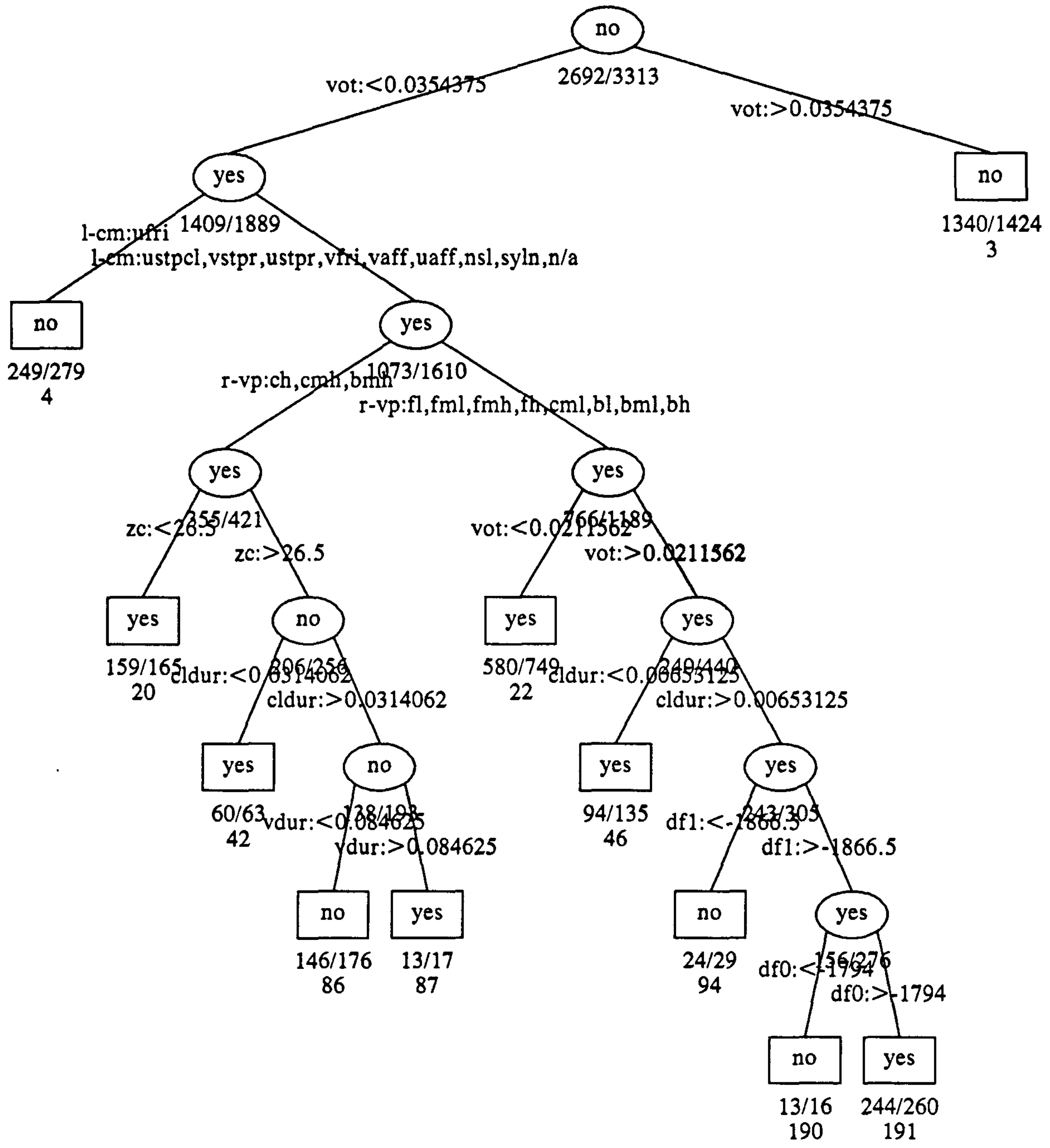


Figure 4. *Full classification tree for voiced vs voiceless stop.*

4. Segment duration modelling for speech synthesis

400 utterances from a single speaker and 4000 utterances from 400 speakers (the TIMIT database) of American English were used separately to build regression trees that predict segment durations based on the following features:

- Segment Context:
  - Segment to predict
  - Segment to left
  - Segment to right
- Stress (0, 1, 2)
- Word Frequency: (rel. 25M AP words)
- Lexical Position:
  - Segment count from start of word
  - Segment count from end of word
  - Vowel count from start of word
  - Vowel count from end of word
- Phrasal Position:
  - Segment count from start of phrase
  - Segment count from end of phrase
  - Segment count from end of phrase
- Dialect: N, S, NE, W, SMid, NMid, NYC, Brat
- Speaking Rate: (rel. to calibration sentences)

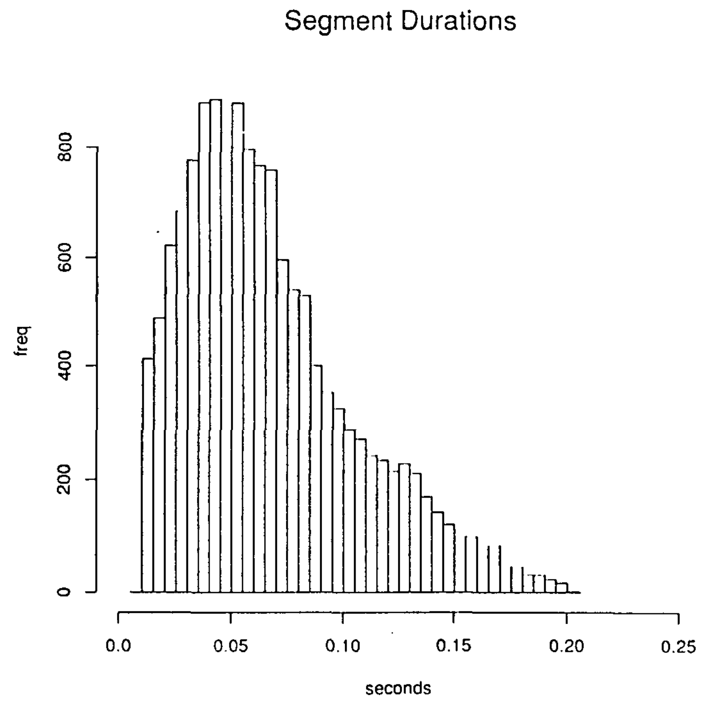
The coding of each segment was decomposed into four features each as described above. The word frequency was included as a crude function word detector and was based on six months of AP news text. The last two features were used only for the multi-speaker database. The stress was obtained from a dictionary (which is easy, but imperfect). The dialect information was coded with the TIMIT database. The speaking rate is specified as the mean duration of the two calibration sentences, which were spoken by every speaker.

Over 70% of the durational variance for the single speaker and over 60% for the multiple speakers were accounted for by these trees. Figure 5 shows durations and duration residuals for all the segments together. Figure 6 shows these broken down into particular phonetic classes. The large tree sizes here, many hundreds of nodes, make them uninteresting to display.

These trees were used to derive durations for a text-to-speech synthesizer and were found to often give more faithful results than the existing heuristically derived duration rules [cf. Klatt 1976]. Since tree building and evaluation is rapid once the data are collected and the candidate features specified, this technique can be readily applied to other feature sets and to other languages.

5. Phoneme-to-phone prediction

The task here is given a phonemic transcription of an utterance, e.g., based on dictionary lookup, predict the phonetic realization produced by a speaker [see also Lucassen, et. al. 1984; Chou, 1987]. For example, when will a T be released or flapped? Figure 7 shows a tree grown to decide this question based on the


Segment Durations
freq
0
200
400
600
800
0.0
0.05
0.10
0.15
0.20
0.25
seconds

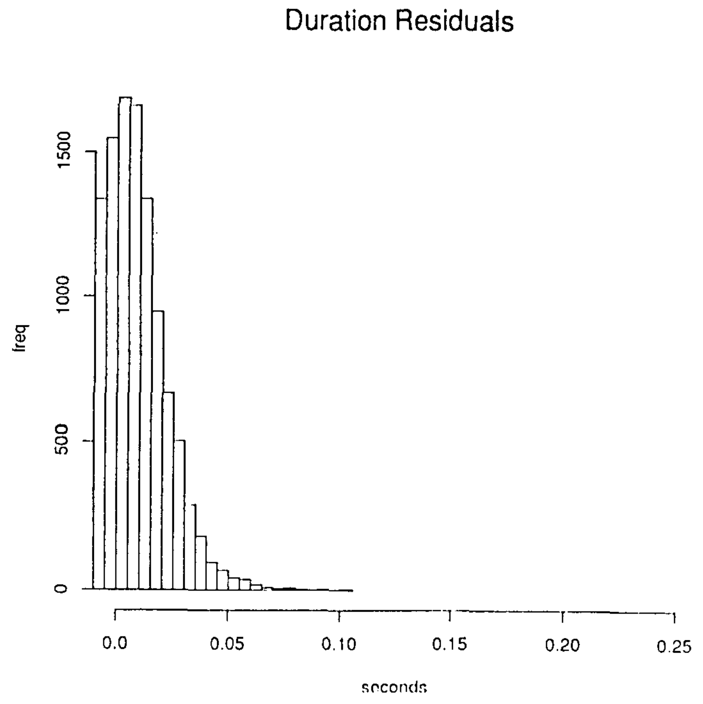
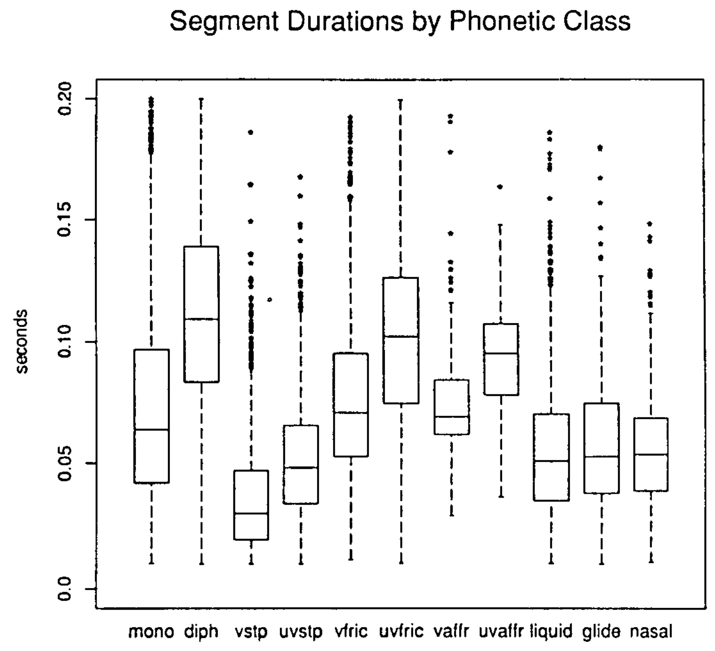
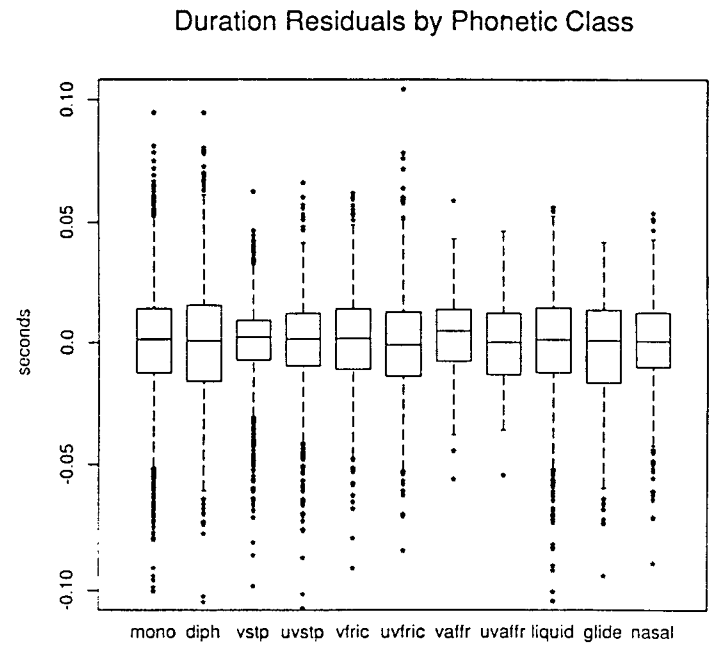

Duration Residuals
freq
0
500
1000
1500
0.0
0.05
0.10
0.15
0.20
0.25
seconds

Figure 5.



Segment Durations by Phonetic Class
seconds
0.0
0.05
0.10
0.15
0.20
mono
diph
vstp
uvstp
vfric
uvfric
vaffr
uvaffr
liquid
glide
nasal



Duration Residuals by Phonetic Class
seconds
-0.10
-0.05
0.0
0.05
0.10
mono
diph
vstp
uvstp
vfric
uvfric
vaffr
uvaffr
liquid
glide
nasal

Figure 6.

TIMIT database. The features used for this tree and a larger tree made for all phones were:

- Phonemic Context:
  - Phoneme to predict
  - Three phonemes to left
  - Three phonemes to right
- Stress (0, 1, 2)
- Word Frequency: (rel. 25M AP words)
- Dialect: N, S, NE, W, SMid, NMid, NYC, Brat
- Lexical Position:
  - Phoneme count from start of word
  - Phoneme count from end of word
- Phonetic Context: phone predicted to left

The phonemic context was coded in a seven segment window centered on the phoneme to realize, again using the 4 feature decomposition described above (and labelled cm-3, cm-2,..., cm3 ; cp-3, cp-2,..., cp3, etc.). The other features are similar to the duration prediction problem. Ignore the last feature, for the moment.

In Figure 7, we can see several cases of how a phonemic T is realized. The first split is roughly whether the segment after the T is a vowel or consonant. If we take the right branch, the next split (to Terminal Node 7) indicates that a nasal, R, or L is almost always unreleased (2090 out 2250 cases) Terminal Node 11 indicates that if the segment preceding the T is a stop or "blank" (beginning of utterance) the T closure is unlabelled, which is the convention adopted by the transcribers. Terminal Node 20 indicates that an intervocalic T preceding an unstressed vowel is often flapped.

This tree predicts about 75% of the phonetic realizations of T correctly. The much larger tree for all phonemes predicts on the average 84% of the TIMIT labellings exactly. A large percentage of the errors are on the precise labelling of reduced vowels as either IX or AX.

A list of alternative phonetic realizations can also be produced from the tree, since the relative frequencies of different phones appearing at a given terminal node can be computed. Figure 8 shows such a listing for the utterance, *Would your name be Tom?* . It indicates, for example, that the D in "would" is most likely uttered as a DCL JH in this context (59% of the time), followed by DCL D (28%). On the average four alternatives per phoneme are sufficient to cover 99% of the possible phonetic realizations. This can be used, for example, to greatly constrain the number of alternatives that must be considered in automatic segmentation when the orthography is known.

These *a priori* probabilities, however, do not take into account the *phonetic* context, only the *phonemic*. For example, if DCL JH is uttered for the D in the example in Figure 7, then Y is most likely deleted and not uttered. However, the overall probability that a Y is uttered in that phonemic context (averaging both D going to DCL JH, D, etc.) is greatest. The point is that to incorporate the fact that "D goes to DCL JH implies Y usually deletes" is that *transition* probabilities should be taken into account.

This can be done by including an additional feature for the phonetic identity of the previous segment. The output listing then becomes a transition matrix for each phoneme. The best path through such a lattice can be found by dynamic programming. This approach would give the best results for an automatic segmenter. This, coupled with a dictionary, can also be used for letter-to-sound rules for a synthesizer (when the entry

Phonetic Realization of T

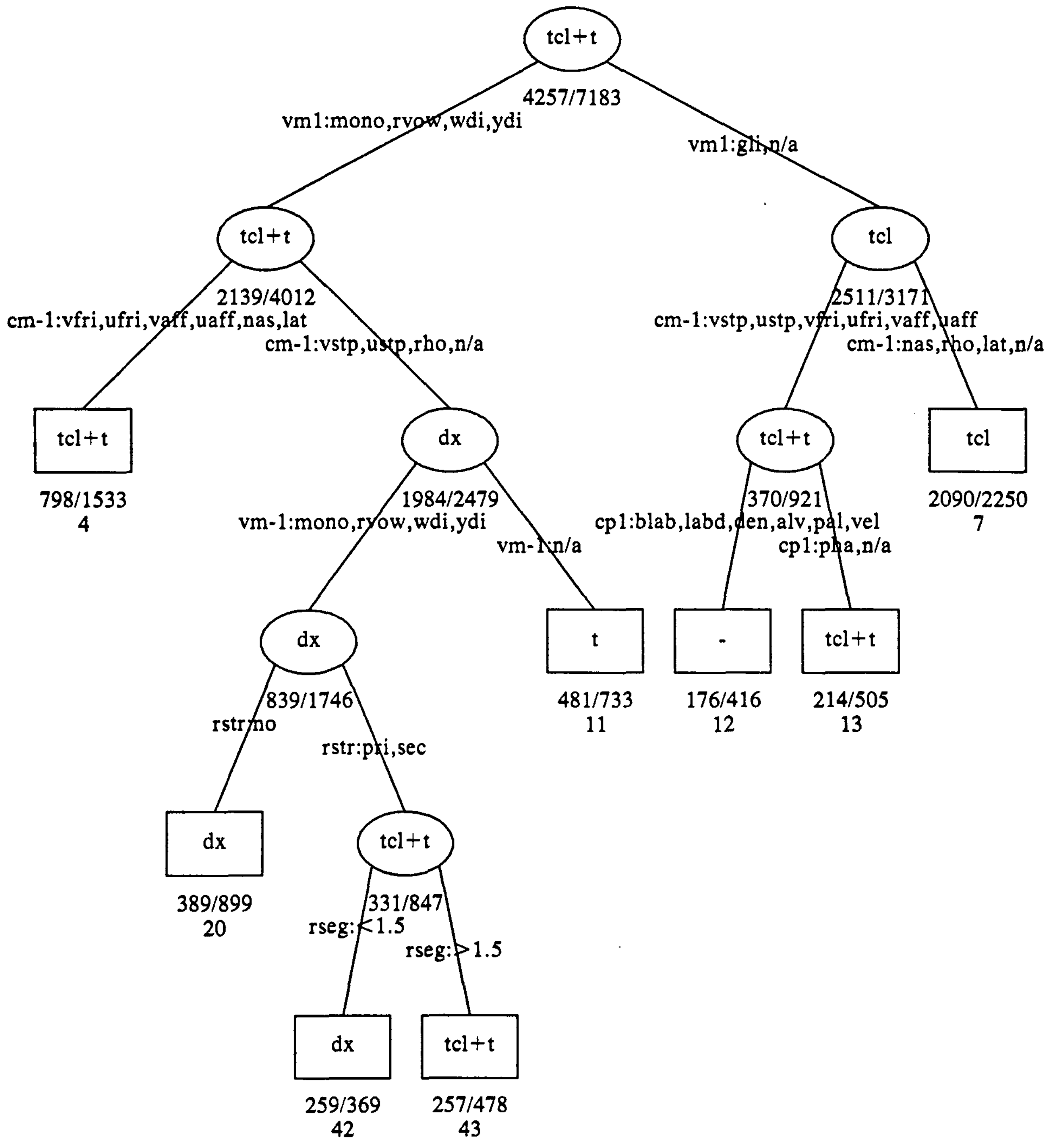


Figure 7. *Classification tree predicting the phonetic realization of phomeme T.*

Would your name be Tom?

Phoneme	Prob	Phone						
w	97.9	w	1.7	-				
uh	79.9	uh	9.2	ix	2.2	uw	2.0	ax
d	59.4	dcl jh	28.1	dcl d	9.4	dcl	3.1	jh
y	76.1	y	22.8	-				
uh	79.9	uh	9.2	ix	2.2	uw	2.0	ax
er	52.6	axr	23.2	r	15.8	er	6.3	-
n	79.8	n	18.6	nx	1.5	-		
ey	95.7	ey	1.3	eh	0.8	ih	0.7	ix
m	96.1	m	3.4	-				
b	87.5	bcl b	4.5	pau b	3.9	bcl	2.5	b
iy	90.5	iy	4.9	ix	2.3	ih	1.2	-
t	92.9	tcl t	5.6	dx	0.6	t		
aa	82.3	aa	7.4	ao	3.4	axr	2.2	ah
m	96.1	m	3.7	-				

Figure 8. *Phonetic alternatives for "Would your name be Tom?*

is present in the dictionary).

The effect of using the TIMIT database for this latter purpose is a somewhat folksy sounding synthesizer. Having the D "Would your" uttered as a JH may be correct for fluent English, but it sounds a bit forced for existing synthesizers. Too much else is wrong! A very carefully uttered database by a professional speaker would give better results for this application of the phoneme-to-phone tree.

6. End of sentence detection

As a final example, consider the not-so-simple problem of deciding when a period in text corresponds to the end of a declarative sentence. While a period, by convention, must occur at the end of a declarative sentence, one can also occur in abbreviations. Abbreviations can also occur at the end of a sentence! The two space rule after an end stop is often ignored and is never present in many text sources (e.g., the AP news).

The tagged Brown corpus of a million words indicates that about 90% of periods occur at the end of sentences, 10% at the end of abbreviations, and about 1/2% in both.

The following features were used to generate a classification tree for this task:

- Prob[word with "." occurs at end of sentence]
- Prob[word after "." occurs at beginning of sentence]

- Length of word with "."
- Length of word after "."
- Case of word with ".": Upper, Lower, Cap, Numbers
- Case of word after ".": Upper, Lower, Cap, Numbers
- Punctuation after "." (if any)
- Abbreviation class of word with ".":
  - — e.g., month name, unit-of-measure, title, address name, etc.

The choice of these features was based on what humans (at least when constrained to looking at a few words around the "."). Facts such as "Is the word after the '.' capitalized?", "Is the word with the '.' a common abbreviation?", "Is the word after the "." likely found at the beginning of a sentence?", etc. can be answered with these features.

The word probabilities indicated above were computed from the 25 million words of AP news, a much larger (and independent) text database. (In fact, these probabilities were for the beginning and end of paragraphs, since these are explicitly marked in the AP, while end of sentences, in general, are not.)

The resulting classification tree correctly identifies whether a word ending in a "." is at the end of a declarative sentence in the Brown corpus with 99.8% accuracy. The majority of the errors are due to difficult cases, e.g. a sentence that ends with "Mrs." or begins with a numeral (it can happen!).

8. References

Bahl, L., et. al. 1987. A tree-based statistical language model for natural language speech recognition. *IBM Researh Report 13112.*

Brieman, L., et. al. 1984. *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks.

Chou, P. 1988. *Applications of information theory to pattern recognition and the desing of decision trees and trellises.* Ph.D. thesis, Stanford University, Stanford, CA.

Klatt, D. 1976. Linguistic uses of segmental duration in English: acoustic and perceptual evidence. *J. Acoust. Soc. Am.* **59**. 1208-1221.

Lucassen, J.M. & Mercer, R.L. 1984. An information theoretic approach to the automatic determination of phonemic baseforms. *Proc. ICASSP '84.* 42.5.1-42.5.4.

Talkin, D. 1987. Speech formant trajectory estimation using dynamic programming with modulated transition costs. *ATT-BL Technical Memo. 11222-87-0715-07.*