From H. Niemann et al., *Progress and Prospects of Speech Research and Technology*, infix, 1994, (pp. 220–232).

# Applications of Decision Tree Methodology in Speech Recognition and Understanding

Roland Kuhn, Ariane Lazarides, Yves Normandin, and Julie Brousseau
Centre de recherche informatique de Montréal
Montréal, Québec, Canada
E-mail: (kuhn OR lazarids OR norman OR broussea)@crim.ca
Elmar Nöth
Bavarian Research Center for
Knowledge Based Systems (FORWISS)
91058 Erlangen–Tennenlohe, F.R. of Germany
E-mail: noeth@forwiss.uni-erlangen.de

**Abstract**

This paper describes decision tree methodology and shows how it has been adapted to three different problems in speech recognition and understanding at CRIM and at FORWISS. The three problems are:

1. Development of context-dependent phone models (work carried out by CRIM researchers and partly inspired by FORWISS work on polyphones). Here, decision trees determine a characterization of the context of a phone that yields good models for recognition.
2. Deriving rules for semantic interpretation from a semantically annotated corpus. This problem led to the development of "Semantic Classification Trees" (in Ph.D. work by R. Kuhn under the supervision of R. De Mori [Kuh93a]).
3. Recognising prosodic features (work carried out in collaboration between FORWISS and CRIM). From a word sequence, Semantic Classification Trees (SCTs) can make predictions about such prosodic features as accents and phrase boundaries. In this ongoing research, we plan to create hybrid decision trees that learn rules combining information from the word sequence and acoustic levels, thus increasing the accuracy with which prosodic events are recognised.

## 1 Introduction

Recent years have seen the development of sophisticated techniques for creating decision trees ([Bre84],[Gel91]). Among the various tools available to pattern recognition researchers, decision trees are remarkable for their robustness and their ability to combine diverse information sources. Furthermore, they give the researcher the freedom to choose which parameters of the problem will be learnt from data, and which will be supplied to the algorithm as *a priori* information.

These characteristics of decision trees make them an ideal tool for solving problems in speech recognition and understanding, which often involve information from multiple levels of the linguistic hierarchy. Much of the collaborative research involving CRIM and FORWISS has focused on decision tree methodology. In this paper, we present applications of the methodology to three very different problems - phonetic modeling, semantic transduction, and prosodic recognition - in order to illustrate just how versatile decision trees can be.

### 1.1 Algorithms for Growing Decision Trees

To grow decision trees, one must supply three elements ([Bre84], the most important single reference for the topic:

1. A set of possible yes-no questions that can be applied to data items;
2. A rule for selecting the best question at any node on the basis of training data;
3. A method for pruning trees to prevent over-training.

The choice of the question set depends entirely on the application (and on the ingenuity of the researcher), while the other two elements tend to be application-independent. Beginning with the root node containing all the training data items, all questions that can be asked at a node are applied to these data items. Each question splits the data into two disjoint subsets whose items return "YES" or "NO" to the question, respectively. The best question is chosen in a way that maximizes the dissimilarity between these two subsets (according to some definition of dissimilarity). The "YES" child of the node inherits the data items that return "YES" to the chosen question, the "NO" child inherits the items that return "NO", and so on recursively. In most applications, the predictive part of a decision tree is located at the leaves. Note that growing a decision tree is a greedy algorithm which does not ensure a global optimum.

It is very important to prevent over-training, which causes a decision tree to "predict" the training data brilliantly, but perform poorly on new data. Many researchers still use the cumbersome cross-validation pruning procedure described in [Bre84]. We use the algorithm described in [Gel91], which is guaranteed to perform as well as or better than cross-validation, is easier to implement, and less computationally expensive. It involves iterative cycles of expansion and pruning on two equal-sized disjoint sets of training data; we will call it the "GRD algorithm" after the three authors of [Gel91]. We highly recommend the GRD algorithm to anyone who grows decision trees!

During the growth of the tree, the set of questions considered as candidates for an interior node may be the same for all nodes, or it may depend on the "history" - the path of "YES" and "NO" answers from the root to the current node. Even if the set of candidate questions is always the same, it may be worth keeping track of the history to avoid asking again a question that was already chosen for an ancestor node, and thus received a "YES" or "NO" answer. In history-dependent applications, the questions at a node are generated from the history; Semantic Classification Trees (described in section 3) illustrate this concept.

## 2 Context-Dependent Phone Models

Consider a phone "ph". The identity of the phones preceding and following "ph" affect the acoustic realization of "ph", so that speech recognition performance can be improved by creating several different models for "ph", each model representing a unique combination of preceding and following phones. In the popular triphone approach, one initially creates a "ph" model for each combination "$< X, ph, Y >$", where $X$ is the preceding and $Y$ the following phone [Lee89]; each such model is called a "triphone". Similar triphones are then clustered until each of the clustered models has sufficient training data.

This approach ignores the strong influence that may be exerted on the acoustic realization of "ph" by phones that are further away than the immediately preceding or immediately following one. FORWISS has developed considerable expertise in this area. FORWISS researchers developed first "context-freezing units"(CFUs) [Sch92] and then "polyphones" [Sch93] as a means of flexibly modeling subword units. CFUs are based on phonologically motivated segments such as syllables, morphemes, and word boundaries, while polyphones allow contexts of arbitrary width and shape. The use of polyphones is made possible by the ingenious APIS training procedure, which exploits the structure of FORWISS's ISADORA speech recognition system [Sch94a, Sch94b].

The FORWISS work stimulated interest at CRIM in going beyond the confines of triphone modeling; the tool chosen for building wider units was the decision tree. Bahl *et al* were the first researchers to apply decision trees to context-dependent phone modeling [Bah91]. The goal is to create a model for every reasonably frequent phonetic context whose average acoustic realization is distinctive from the average acoustic realization for other contexts. Suppose that we have acoustic data for "ph" in a variety of phonetic contexts, a method for training probabilistic models for acoustic realizations of a phone, and a measure of the distance between two such probabilistic models. We can now grow a decision tree in which the candidate questions at a node concern the phonetic context of "ph". The question chosen for the node is the one in which the combination of the "YES" model for "ph" (trained on data items yielding "YES" to the question) and the "NO" model (trained on data items yielding "NO") yields the greatest improvement over the model at the node itself. Let the phone preceding "ph" be denoted "$-1$", the phone before that "$-2$", and so on; similarly, the phone after "ph" can be denoted "$+1$", the one after that "$+2$", and so on. A typical question in the resulting tree might be: "Is $-2$ an 's'?" Each leaf of the tree corresponds to a context-dependent phone model.

In an HMM-based speech recognition system, the obvious type of probabilistic model to use for evaluating questions is an HMM. However, if one wishes to consider a large question set, HMMs are computationally too expensive. Bahl *et al* devised a computationally cheap Poisson model for the acoustic data for "ph" in a given context. This model allows extremely fast question evaluation; by using it, these researchers were able to consider questions involving all possible phones at the five preceding and five following positions. Of course, the models found at the leaves of their trees are conventional HMMs.

We have extended this work in four ways:

1. We use both the Poisson and the HMM MLE criteria;
2. We prune an over-large tree instead of applying stopping criteria;
3. Our trees are ternary instead of binary;
4. As well as asking about phones, we have experimented with questions involving phonetic features.

## 2.1 Question Scoring: Poisson and HMM MLE Criteria

The Poisson criterion can be evaluated quickly, but is a very rough approximation - for instance, it ignores the evolution of the signal for "ph" over time. An HMM-based criterion is a better, if computationally expensive, way of choosing the best question at each node of the tree. We employ the Poisson criterion to find quickly the $M$ best questions at a node during tree expansion, then use an MLE criterion applied to HMMs to make the final choice from these. Tree pruning (see the next point) is carried out with the same HMM-based criterion.

## 2.2 Pruning vs. Stopping Criteria

Bahl *et al* use stopping criteria to halt the growth of a tree; that is, instead of pruning an over-large tree, constraints are applied during tree growth. A node is turned into a leaf node if the score of the best question falls below a threshold, or if the number of data items is too small. Unfortunately, the decision tree literature is unanimous in maintaining that this is a poor method for avoiding over-training: an apparently bad question (turned into a leaf node by the stopping criteria) might have yielded descendants that perform well [Bre84], [Gel91]. Thus it is better to grow an over-large tree, then prune it back by examining its performance on new data. By modifying the iterative GRD expansion-pruning algorithm [Gel91], we were able to grow and prune phone context trees in a more suitable way.

## 2.3 Ternary Trees and Search Algorithms

Our decision trees are ternary rather than binary - each interior node has "YES", "NO", and "UNKNOWN" children, so that the tree contains models to be used in every conceivable combination of knowledge and ignorance about the phonetic context. This permits the trees to support an almost unlimited range of search algorithms. Like Bahl *et al*, we allow questions about the five preceding and five following phones. However, a given search algorithm may make some of these questions difficult to answer.

Suppose that the identity of the +2 phone and following + phones is currently unknown. If the question at the root of the tree happens to concern the +3 phone, we proceed to the subtree at the UNKNOWN child of the root, which is guaranteed to contain no questions concerning the +3 phone or phones further out. If the first question we encounter concerns the +2 phone, we can again enter the UNKNOWN subtree and recurse until we encounter a suitable question (for instance, one concerning the −1 position, if we know about that position). This kind of tree is well-suited to multi-pass search, in which the first few passes score paths in the graph on the basis of incomplete information, while the last pass uses knowledge about the complete context of current phone "ph".

During tree-growing, some of the data items at a parent node are copied to the YES child and the remainder are copied to the NO child, but the UNKNOWN child inherits **all** data found in the parent. If the question chosen at the parent involves position X, the questions considered for the UNKNOWN child are exactly those considered at the parent, minus those involving position X or positions further out. Note that the leaf node attained by traversing the path made up only of "UNKNOWN" answers contains the model appropriate when we know nothing about the context of "ph" - that is, a model trained on **all** examples of "ph" in the training data.

## 2.4 Features

Most work on phone context modeling considers the context of a phone as also being defined in terms of phones. However, it is quite possible that some articulatory traits in the phones near the phone "ph" being modeled have a stronger impact on the acoustic realization of "ph" than others. It might be very important to know whether the −1 phone is a nasal or not, but irrelevant whether it is 'm' or 'n'. We experimented with two feature schemas for generating the set of candidate questions at a position. Schema 1 is based on the Chomsky-Hall feature definitions, schema 2 is based on articulatory feature definitions [Lad82]. Since the units currently in use at CRIM were not based on either schema (they include units for "silence" and for "pause", and several dipthongs) the definitions used for our experiments do not correspond exactly with those used in [Lad82]. We plan to carry out further experiments in which the units and the feature set match their precise definitions in the linguistic literature. We also plan experiments in which both questions about phones and questions about features may be considered, so that (for instance) a leaf node might model occurrences of "ph" in which −1 is 's' and +1 is voiced.

Preliminary results for this work can be found in [Nor94]. Figures $1-3$ show trees obtained for the phone "uh": Figure 1 shows a tree with phone-based questions, while figures 2 and 3 show trees obtained using feature schemas 1 and 2 respectively. In figure 1, note that if the answers to questions about the $+1$ and $-1$ positions are both "UNKNOWN", we assume that information further out is unavailable and we arrive at a leaf node. If the current node is descended from a node in which a question about a position was answered "YES" or "NO", it is forbidden to answer "UNKNOWN" to a question about that position or a position further

## 2.6 Future Work

- We plan to carry out experiments in which both questions about phones and questions about features are considered. It is possible, for instance, that in the $-1$ and $+1$ positions it is often important to know the precise identity of the phone, while at more distant positions knowledge about one or two features suffices.

- In some recent work on triphones, tree-based clustering is carried out for each state of an HMM, rather than for the whole HMM [You94]. Our algorithms could be applied here with only minor changes; we plan to carry out the appropriate experiments soon.

- Bahl *et al*'s criterion could be described as a Poisson MLE criterion. We have devised a Maximum Mutual Information Estimation (MMIE) variant of their criterion that employs the same Poisson model. With this, it would be possible to choose questions in the tree for "ph" in a way that lowers the risk that "ph" will be confused with other phones, which is precisely the criterion desired for speech recognition. The Poisson MMIE criterion will be computationally more expensive than the Bahl *et al* Poisson MLE, but considerably less expensive than HMM MMIE (which is impractical for use in phonetic context modeling). The really interesting comparison will be between Poisson MMIE and HMM MLE - if we are lucky, the former might yield better recognition results than the latter, yet be computationally less expensive.

# 3 Semantic Classification Trees (SCTs)

## 3.1 Introduction

The **Semantic Classification Tree** is a specialized type of decision tree that learns semantic rules from training data and can be a building block for natural language understanding (NLU) systems [Kuh93a], [Kuh93b], [Mil93]. By reducing the need for handcoding and debugging a large number of rules, SCTs facilitate rapid construction of an NLU system. SCTs are particularly well-suited to speech understanding - they are highly resistant to errors by the speaker or by the speech recognizer because they depend on a small number of words in each utterance. Though space does not allow us to describe it here, our work on the ATIS task showed that SCTs can yield successful NLU for a realistic application [Kuh93b].

SCTs have the following properties:

- They learn rules for classifying new strings or substrings from a corpus of classified strings or substrings. To apply SCTs to a problem, one must formulate it as a classification problem.

- The questions in the nodes of an SCT involve regular expressions made up of string symbols and a special gap symbol. The string symbols could be words or higher-level constituents.

- Generation and selection of questions is completely automatic: any symbol from the symbol lexicon may appear in a question.

Compared with other types of decision trees, the original aspect of SCTs is the way in which the set of possible questions is generated. These questions ask whether a word sequence matches certain regular expressions involving words and gaps. To choose a question from this set, we use the Gini "impurity" $i(T)$ of a node $T$ [Bre84].

Figure 4 shows an example of a single-symbol SCT grown on sentences from the ATIS domain. Its job is to decide whether a request should result in showing the user the attribute *fare.fare_id* (found in an air-travel database) or not; sentences that end up in a YES leaf will have *fare.fare_id* in their "displayed attribute list", used in generation of an SQL query. The symbols "$<$" and "$>$" match the start and end of a sentence, a "+" between two words or symbols indicates a gap of at least one word between them, and the expression "$M(w)$" (e.g. "$M(fares)$" in the figure) matches one or more occurrences of the word $w$; order matters. For instance, the input "Show me first-class fare flights to Boston" matches the pattern $< +fare+ >$ at the root, does not match $< +fare\ code+ >$, and does match the pattern $< +fare\ flights+ >$ - it thus yields "NO". "Show me the fare for flights to Boston" matches the root expression but no other expression it encounters, and thus yields "YES" (it does not match $< +fare\ flights+ >$ because "for" comes between "fare" and "flights").

## 3.2 Growing an SCT

Each node of the growing single-symbol SCT is associated with a regular expression called the *Known Structure* consisting of *symbols* and *gaps* (denoted "+"); the set of possible questions is generated by operations on the gaps. The known structure for the root of the SCT is $< + >$; strings entering the root must have length at least one.

The four expressions generated from a given gap + in the known structure $KS$ and a given lexical item $w_i$ are:

1. The expression obtained by replacing + in $KS$ by $w_i$;

2. The expression obtained by replacing + in $KS$ by $w_i+$;

3. The expression obtained by replacing + in $KS$ by $+w_i$;

4. The expression obtained by replacing + in $KS$ by $+w_i+$.

At the root, whose known structure is $< + >$, these four *gap operations* generate the expressions $< w_i >$, $< w_i+ >$, $< +w_i >$, and $< +w_i+ >$. Each of these expressions $E$ is turned into a potential question by asking: "Does the sequence being classified match the expression $E$?" If there are $V$ symbols in the lexicon, $4 * V$ questions are generated by allowing $w_i$ to be any of them. ¿From these, the algorithm selects the one which achieves the best split of the training data. As the tree grows, known structures get longer. If the question "does the sequence match $< +w_8+ >$?" is selected to fill the root, the known structure for the root's YES child is $< +w_8+ >$, and the known structure for the root's NO child is $< + >$. New questions are generated by applying the four gap operations to each + individually.

### 3.3 Classifying Substrings

The algorithms for growing SCTs can be adapted to the task of classifying substrings. An example from ATIS will illustrate what we mean by this. Consider the sentence "show me flights from Boston no sorry from New York to Chicago stopping over in Pittsburgh". After parsing, this would be "show me flights from CIT no sorry from CIT to CIT stopping over in CIT". The CITs should be labelled as follows: "show me flights from CIT⇐SCRAP no sorry from CIT⇐ORI to CIT⇐DEST stopping over in CIT⇐STOP", where "ORI" is flight origin, "DEST" is destination, "STOP" is stopover, and "SCRAP" means the CIT is irrelevant.

Work carried out at FORWISS verified the good results achieved with SCTs in the ATIS domain for the German train inquiry domain (see 4.2 for a description of the data used for these experiments). Using the best word sequence hypothesis of the recogniser, we classified city names in the hypothesis into the four semantic concepts "origin", "stopover", "destination" and "recognition error". On average, an utterance contained 1.5 city names (i.e. substrings) to be classified. The sentence accuracy was 40%. In 97% of the utterances every city name was correctly classified into one of the four concepts.

The SCT-growing algorithm described above requires only minor modification to grow SCTs that classify parts of strings. The key is to submit the same sentence to an SCT as many times as there are substrings to be classified, each time "marking" the substring to be classified with a special symbol.

## 4 Applications of SCTs to Prosodic Recognition

### 4.1 Introduction

When we listen to someone speak, we often perceive boundaries between phrases, accent being laid on particular words in a sentence, and other prosodic events. These disambiguate our understanding of the speaker's meaning, and may even help us to distinguish between similar-sounding words. Apart from the intrinsic linguistic interest of prosodic phenomena, they have practical value. For instance, a speech recognition system that detected them with high accuracy could use this information to rescore N-best sentence hypotheses [Wan92]. FORWISS researchers and their colleagues have shown that "Prosodically marked Clause Boundaries" (PCBs) can guide the construction of a parse tree for a sentence, yielding a 70% reduction in the number of syntactic derivations [Bak94], and that subword units whose models incorporate lexical accent information reduce word and sentence recognition error rates by about 5% [Bat93].

However, prosodic recognition is a hard problem. As with speech recognition, the solution seems to lie in combining high-level linguistic knowledge (analogous to language models for speech recognition) with knowledge about acoustics [Bak94], [Bat93], [Kom94]. The motivation for the current FORWISS-CRIM collaborative work on prosody is to employ SCTs to predict prosodic events based on the recognized word sequence. Since decision trees make it easy to merge information from dissimilar sources, our long-term goal is to grow decision trees that use both SCT-style questions and questions about the acoustic signal to recognise prosodic events.

### 4.2 The Experimental Setting

In their prosodic recognition work, the FORWISS researchers adopted the statistical paradigm; thus, they required a large training database consisting of sentences with reference labels for prosodic phenomena of interest (phrase boundaries and lexical accents). A stochastic context-free sentence generator yielded a text corpus of 10,000 sentences in the ERBA (Erlangen train enquiry) domain [Bat93]. These sentences, containing a vocabulary of 949 words (including 571 train stops) were then recorded by 100 untrained speakers in a quiet office environment and partitioned into training and testing subcorpora. The generating grammar gave expected prosodic phrase boundaries within sentences. The position of these was based on syntax and prosodic knowledge (see below, B1). They were **not** included in the version of the sentences given to the speakers. Similarly, based on syntactic structure and also on information about word accents in the lexicon, the expected lexical accents within each sentence could be obtained.

There are four types of automatically generated phrase boundaries:

- **B3**: boundaries between elliptic clause and clause, between main and subordinate clause, or at coordinating particles between clauses;
- **B2**: boundaries between constituents, and boundaries at coordinating particles between constituents;
- **B1**: boundaries of **B2** type that are expected not to be prosodically marked, because they are close to a **B3** boundary or the beginning or end of the utterance;
- **B0**: all other boundaries between words.

To verify these expectations about prosodic marking of these phrase boundaries, perception experiments were run on 500 utterances [Bat93]. Each utterance was played to ten naive listeners, who were asked to mark perceived boundaries between words. In 93% of the cases where at least 6 listeners perceived a boundary, **B2** or **B3** had been automatically generated; in 90% of cases where less than 6 listeners perceived a boundary, **B0** or **B1** had been automatically generated. It was concluded that the automatically generated boundaries reflect human perceptions quite well, and that they may be used to train prosodic classifiers.

## 4.3 Designing a Prosodic Classifier

The first prosodic classifiers designed by FORWISS researchers and their collaborators relied entirely on information contained in the acoustic signal (a preliminary description is in [Bat93], a more up-to-date one in [Kom94]). After the signal had been automatically time-aligned with the word sequence, 31 acoustic parameters were extracted for each word boundary; these included the length of the pause, average speaking rate over the whole sentence, duration- and energy-related parameters for syllables surrounding the boundary, and parameters describing the shape of the F0-contour covering the boundary.

Three types of classifiers for distinguishing boundary types were trained on these data: polynomial classifiers, Gaussian distribution classifiers, and artificial neural networks (ANN). Performance results obtained on test data from perception experiments are given in [Kom94, Kie94]; the ANN performed best. Further significant improvement was obtained by applying a simple stochastic language model which looks at surrounding words during search to rescore the output of the acoustic classifier.

By contrast, another group of researchers relied entirely on word sequence information to predict prosodic boundaries; no acoustic information whatsoever was used [Wan92]. Interestingly, their recognition rates were in the same range as those of the FORWISS researchers and their collaborators, who relied mainly on acoustic information. The integration of these two approaches seems likely to yield a notably better performance than either approach on its own.

## 4.4 Why SCTs?

Section 3 above describes Semantic Classification Trees (SCTs), which learn rules for assigning classes to sentences or parts of sentences. First experiments with the n-best word chains show that both capacities are useful in the context of ERBA. For instance, one could classify sentences in this domain:

```
Ich will nach Muenchen => FAHRPLANAUSKUNFT
(I want to go to Munich => TRAVEL_PLAN_INFORMATION)

Was kostet die Rueckfahrkarte => FAHRPREISAUSKUNFT
(How much does the return ticket cost => TRAVEL_FARE_INFORMATION)
```

For prosodic recognition, however, the ability to classify substrings is of more interest. If we are interested in classifying boundaries, all we need to do is insert a special symbol such as "|" between each pair of words. The SCT must then assign a boundary class to each word boundary "substring". Consider the utterance

```
Good B0 morning B3 I B1 would B0 like B1 a B0 train B2 to B0 Munich
```

This utterance gives 8 training items that would then look like this (the "*" marks the boundary to which the type after the arrow applies):

```
Good *| morning | I | would | like | a | train | to | Munich => B0
Good | morning *| I | would | like | a | train | to | Munich => B3
...
```

What are the advantages of using SCTs, rather than a conventional $N$-gram approach, to predict prosodic boundary types?

- SCTs can model long distance dependencies - if prosody is influenced by such factors as the topic of a sentence or the mood of the speaker, as reflected in words distant from the boundary in question, an SCT can learn the appropriate long-distance rules.
- SCTs tolerate many insertion, deletion, and substitution recognition errors - if they are trained on error-containing output from the recognizer itself (e.g. the $N$-best hypotheses) they become even more robust.
- Once grown, SCTs are extremely fast classifiers - if appropriate, they can be used to speed up search (for instance by controlling the stack of a conventional parser).
- Preliminary experiments (see next subsection) show high accuracy for SCT-based prediction of phrase boundary type in the ERBA domain - performance is very good even when the SCTs are trained on relatively small amounts of data.
- The decision tree methodology lends itself to integration of diverse information sources - we plan to design enhanced SCTs that ask questions about acoustic parameters, as well as about the word sequence.

## 4.5 Preliminary Experiments

Our first experiment was intended to see how well we can predict the boundaries based on purely textual information. We started with the spoken word chain and tried to classify the three classes B01, B2, and B3 (the *a priori* probabilities of the classes are 76, 21, and 3% respectively). It turns out that even a small training set yields impressive results: with a training set of 100 sentences ($\approx$ 1000 word boundaries) recognition rates of 97 (B01), 78 (B2), 58 (B3), and 91% (total) were achieved (learn $\neq$ test). An obvious problem is the large discrepancy between the *a priori* probabilities of the three classes which results in bad recognition rates for small classes. Training with a class distribution somewhere between equal and *a priori* probability can thus give better results: we took 4100 boundaries from 2000 sentences for training (2000 B01, 1400 B2, and 700 B3) and achieved 91 (B01), 91 (B2), 98 (B3), and 92% (total) (again learn $\neq$ test).

Given that the syntactic structure of the corpus is limited due to the way it was constructed (which means that the recognition rates are higher than to be expected in “real life”), it has to be noted that the training set was relatively small. An analysis of the selected key words (a total of 78) showed the ability of the SCT to generalize: 40% of the words were function words. The content words were verbs or nouns typical for the application (like “to go”, “time of arrival”). There were also key words like city names, indicating that the training data were not enough to generalize. Thus the recognition on the test set can be further improved.

### 4.6 Future Work

The work described above is merely a beginning. There are several avenues for further exploration:

- We plan to investigate the integration of SCT-style rules with rules based on the acoustic features. This could be done by supplying the tree-growing algorithm with a repertoire of both types of question, permitting it to select whichever type provides the best “split” of training data at a node. A slight problem is raised by the continuous nature of acoustic parameters, since decision tree questions must be “yes-no”. In such situations, it is customary to supply questions of the form “Is $x_i < p_j$?”, where $x_i$ is the value of the parameter for the data item and $p_j$ a constant supplied by the researcher [Bre84]. Thus, we would have to determine an appropriate set of $p_j$ for each acoustic parameter.

- Experiments must be done to determine the optimal set of acoustic parameters.

- Instead of yielding a **class**, the trees should yield a **weight** or **probability** for each possible type.

- Prediction of lexical accents as well as of prosodic boundaries should be investigated.

- Once we have good prosodic recognition, we plan to incorporate it into our general speech recognition and understanding systems (to improve subword modeling and syntactic parsing).

- Finally, we hope to bootstrap our work on predicting prosodic phenomena for an automatically generated corpus into a system that can predict these phenomena in spontaneous speech.

## 5 Acknowledgements

This paper owes its existence to the CRIM-FORWISS collaboration, funded jointly by the Ministère des Affaires Internationales du Québec, the government of Bavaria and the German Ministry for Research and Technology (*BMFT*) through the International Bureau of the GKSS Research Centre (*GKSS/IB*). The CRIM portion of the work was conducted in collaboration with CAE Electronics, CRIQ, and IREQ, as part of a research effort partly sponsored by the Canadian Ministry of Industry. The authors of this paper found the opportunity to exchange ideas with researchers working on related problems, but in a different country and a different research milieu, invaluable.

## References

[Bah91] L. R. Bahl, P. V. de Souza, *et al*, “Decision Trees for Phonological Rules in Continuous Speech”, *ICASSP-91*, V. 1, pp. 185-188, 1991.

[Bak94] G. Bakenecker, U. Block, *et al*, "Improving Parsing by Incorporating 'Prosodic Clause Boundaries' Into a Grammar", to appear in *ICSLP-94*, 1994.

[Bat93] A. Batliner, R. Kompe, *et al*, "The Prosodic Marking of Phrase Boundaries: Expectations and Results", *Proc. NATO ASI Conference 1993*, V. 2, pp. 89-92, Bubion, Spain, 1993.

[Bre84] L. Breiman, J. Friedman, *et al*, "Classification and Regression Trees", Wadsworth Inc., 1984.

[Gel91] S. Gelfand, C. Ravishankar, and E. Delp, "An Iterative Growing and Pruning Algorithm for Classification Tree Design", *IEEE Trans. PAMI*, V. 13, no. 2, pp. 163-174, Feb. 1991.

[Kie94] A. Kießling, R. Kompe, *et al*, "Detection of Phrase Boundaries and Accents", in this volume, Sept. 1994.

[Kom94] R. Kompe, A. Batliner, *et al*, "Automatic Classification of Prosodically Marked Phrase Boundaries in German", *ICASSP-94*, V. 2, pp. 173-176, Adelaide, Australia, 1994.

[Kuh93a] R. Kuhn, "Keyword Classification Trees for Speech Understanding Systems", *Ph.D. Thesis*, McGill University, June 1993.

[Kuh93b] R. Kuhn and R. De Mori, "Learning Speech Semantics with String Classification Trees", *ICASSP 93*, V. II, pp. 55-58, Minneapolis, Apr. 1993.

[Lad82] P. Ladefoged, "A Course in Phonetics", Harcourt Brace Javanovich, 1982.

[Lee89] K.-F. Lee, "Automatic Speech Recognition - the Development of the SPHINX System", Kluwer Academic Publishers, 1989.

[Mil93] E. Millien and R. Kuhn, "A Robust Analyzer for Spoken Language Understanding", *Eurospeech 93*, V. II, pp. 1331-1334, Berlin, Germany, Sept. 1993.

[Nor94] Y. Normandin, R. Kuhn, *et al*, "Recent Developments in Large Vocabulary Continuous Speech Recognition at CRIM", in this volume, Sept. 1994.

[Sch92] E. G. Schukat-Talamazzini, H. Niemann, *et al*, "Acoustic Modelling of Subword Units in the ISADORA Speech Recognizer", *ICASSP-92*, pp. 577-580, 1992.

[Sch93] E.G. Schukat-Talamazzini, H. Niemann, *et al*, "Automatic Speech Recognition without Phonemes", *Eurospeech 93*, V. I, pp. 129–132, Berlin, 1993.

[Sch94a] E.G. Schukat-Talamazzini, "Speech Recognition for Spoken Dialog Systems", in this volume, Sept. 1994.

[Sch94b] E.G. Schukat-Talamazzini, "Automatische Spracherkennung", Vieweg, Wiesbaden, 1994 (to appear).

[You94] S. J. Young, J. Odell, and P. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modelling", *ARPA Workshop on Human Language Technology*, pp. 286-291, Mar. 1994.

[Wan92] M. Wang and J. Hirschberg, "Automatic Classification of Intonational Phrase Boundaries", *Computer Speech and Language*, 6(2), pp. 175-196, 1992.