EXPERIMENTS WITH AN OBJECT-ORIENTED SPEECH DATABASE

M. Vainio[1], T. Altosaar[2], M. Karjalainen[2] and Antti Iivonen[1]
[1]University of Helsinki, Dept. Phonetics
P.O. Box 35 (Vironkatu 1 B), 00014 Helsinki, Finland
[2]Helsinki University of Technology, Acoustics Lab.
Otakaari 5 A, 02150 Espoo, Finland

ABSTRACT

This paper describes a very flexible environment in which to perform speech related experiments along with some examples. A complete range of operations for the speech researcher is available from low-level signal processing to high-level phonetic analyses and semi-automatic transcription using neural networks. Entities such as signals and transcriptions are all represented as objects and can be stored between sessions via a transparent database system.

INTRODUCTION

Speech databases have been developed for the major European languages and have been available for use for several years already [1]. However, the lack of a comprehensive and transcribed speech database for Finnish has hindered research in speech analysis and recognition for this language. In 1991 we initiated a project in which Finnish speech material was collected into a database and transcribed manually by phoneticians [2].

Since we desired a very flexible environment in which to perform speech experiments we decided to implement the database on top of our object-oriented QuickSig signal processing system [3]. The entire system is written in Common Lisp and CLOS and provides for a seamless integration of all activities. The system includes input and output audio channels, graphical tools for the user to move around in the database, and transcription frames with semi-automatic transcription aids, such as diphone detectors implemented with neural networks. Loading, caching, and storing of signals is managed by the system automatically and transparent to the user. An advanced speech representation framework is used to represent phonetic and linguistic information and can be used in speech processing tasks such as analysis, synthesis and recognition. The framework allows for abstract, structural, specific, and fuzzy phonetic objects to exist over different scales, e.g. from sentences down to acoustic segments. Transcriptions are automatically transformed into these linked speech representation objects when accessed from persistent store.

Predicate functions can be designed and applied to search over all or part of the database. For example, a search over part of the database that includes 20,000 transcribed phonemes can be performed in a few seconds. The search returns the phonetic objects that matched the predicate, e.g., a set of phonemes. These phoneme objects are all linked to their original signals and thus can be used in a wide variety of signal processing methods and techniques found in QuickSig such as spectral averages calculated over specified regions of the speech signal, formant analysis, and duration analysis.

LABELING PROCESS

Labeling of speech signals is accomplished with the aid of transcription frames. The transcription frame serves as a graphical user interface between the technical aspects of the transcription process and the transcriber. Any number of different transcriptions may be linked to a signal since different interpretations may be required in some situations.

Segmentation

To ensure reliable and consistent segmentation it is important that the user can utilize decision-aiding transcription tools. Conventional tools include time-waveform, FFT-spectrogram, and energy contour displays.

In addition to these standard tools we have implemented more advanced methods that rely on auditory modeling. An auditory spectrogram on the Bark scale and a loudness contour usually permit more accurate segmentations to be made. A spectral change measure calculated from auditory spectra covering the span of the signal often indicate fairly accurate locations for segment boundaries.

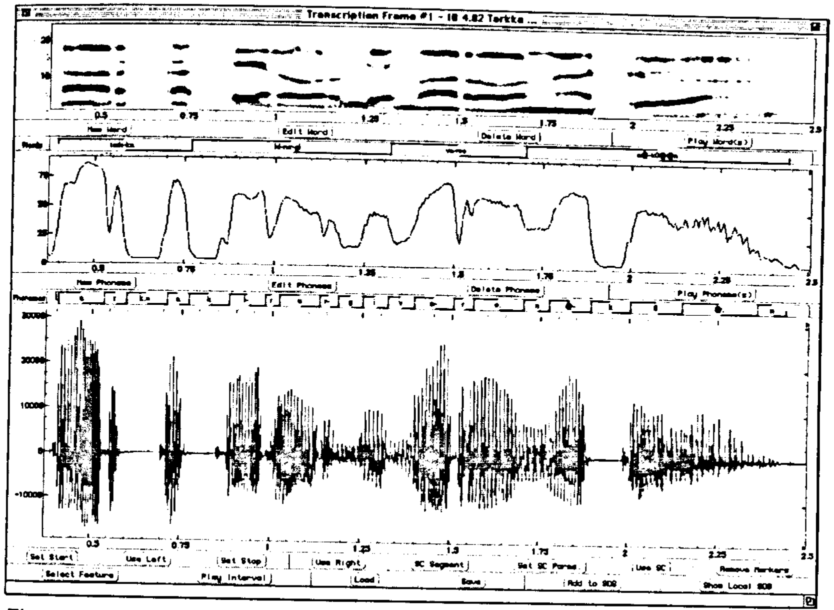


Figure 1. Transcription frame for labeling speech signals. An auditory spectrogram, a loudness contour, and the time waveform are shown along with a word and phonetic level transcriptions.

Neural networks can also be used to determine segment boundaries. Diphone detecting networks can offer reliable semi-automatic hypotheses for phoneme boundaries.

Labeling

Signals are usually segmented according to their structure on a phonemic level. In addition to this phonemic representations it is possible to label material in several hierarchical levels ranging from low-level acoustical segments to syllables, words, and sentences.

The user can define the symbols used for labeling operations according to specific needs. For instance, a more narrow transcription may be required in some cases. This may be accomplished by adding a new representational level to the existing transcription, or, by creating an alternative transcription altogether.

Reading of other transcribed material from databases is also possible and can be viewed in the transcription frame.

Figure 1 shows a transcription frame for the Finnish sentence <tarkka kirurgi varoo näköään>. Besides the time-waveform, loudness contour, and auditory spectrogram, two different levels of transcription are visible: a phoneme and a word level. Frequently used operations have been assigned their own buttons and allow the user to perform different functions such as playing portions of the signal, assigning boundaries, and invoking different transcription aids.

DATABASE

The database system has also been implemented in an object-oriented programming fashion and is designed for simple and user transparent operation. Besides containing signals, the database can store transcriptions, speaker related information, and in general any user designed object. Links between different objects allow for deferred loading, i.e., an object is loaded into memory only when required for a calculation and discarded when not needed. This means that an entire speech database can exist in working memory simultaneously. Fast and efficient analyses can thus be performed on large amounts of material.

Objects in the database are arranged in a hierarchical manner. When a user terminates a session the database system automatically checks for objects that have been created or changed and are transferred to permanent store.

Figure 2 shows a graphical representation of part of a Finnish speech database. Nodes in the tree are mouse-sensitive and allow for different operations to be performed, e.g., opening a transcription frame, playing signals, and inspecting the state of specific objects.

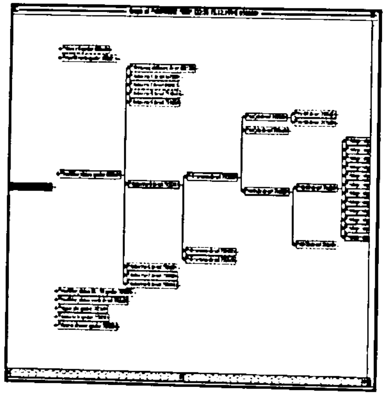


Figure 2. A graphical representation of part of a speech database. By default nodes are collapsed but may be opened with the mouse.

The database currently contains approximately two hours of labeled speech from two male and two female native Finnish speakers. At this stage most of the database consists of isolated words but sentence material is being added.

PHONETIC HIERARCHY

All of the phoneme symbols created during labeling are transformed into instances that have a implicit feature structure. Figure 3 shows part of this network seen from a phonological viewpoint. The use of object-oriented CLOS class hierarchies define the specific structural relations between different phonetic units. These relations can be used to represent phonetic and linguistic information when performing searches and analyses over the database.

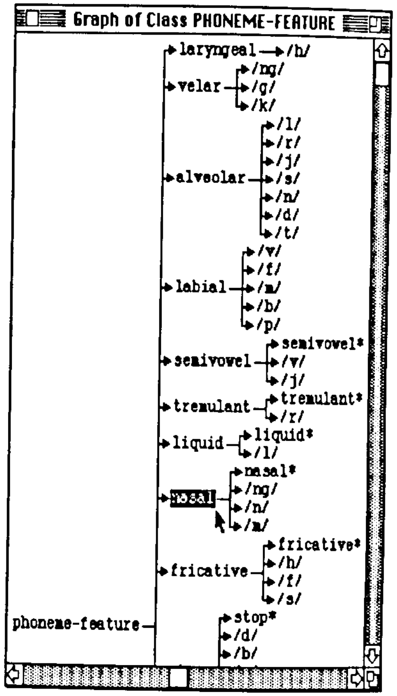


Figure 3. Part of the phoneme inheritance network graphed according to phonological features.

ANALYSES

In this section we briefly present some of the methods that are available for advanced phonetic analyses.

Duration Analysis

In this example the duration distribution of the vowel /i/ in a C/i/C context is to be calculated for a single speaker. First, a predicate is defined using Lisp syntax:

```
(define-predicate C/i/C
 (and
  (previous-phoneme is-not-a V)
  (x is-a /i/)
  (next-phoneme is-not-a V)
  (speaker-is "MV")))
```

Then a search over the entire database is performed and a set of phonemes matching the predicate is returned. Each phoneme object has its own internal state and can determine its duration in time. A histogram can be built from these phonemes and is shown in figure 4.

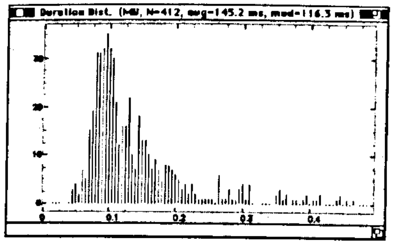


Figure 4. Histogram of the vowel /i/ in a consonant context (x-axis represents time in seconds).

Spectral Average

Each phoneme not only knows it duration but also its absolute position in time. This information may then be used by any analysis method. One such interesting analysis is to calculate the auditory spectrum at the midpoint of each phoneme's signal span. The average spectrum and the spectral distribution for all 412 /i/ vowels found in the previous example are shown in figure 5.

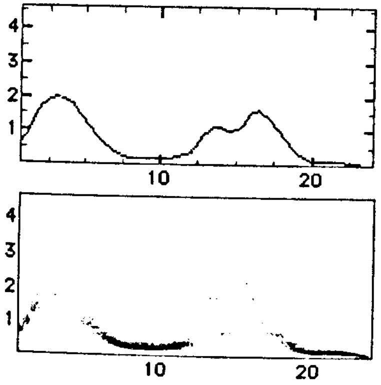
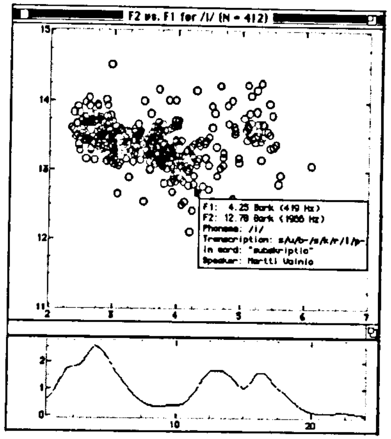


Figure 5. Auditory spectral average (top) and distribution (bottom) for 412 /i/ vowels (amplitude vs. frequency on the Bark scale).

Formant Analysis

Each individual phoneme's auditory spectrum can also be automatically analyzed for formant locations which can then be displayed on a F1/F2 chart. In figure 6 the same 412 /i/ vowels have been analyzed and displayed. Individual vowels have their formants represented by circles which are mouse sensitive. This allows the user to inspect each item separately and access information such as exact formant locations, the transcription and word in which the phoneme is situated, and other related information such as speaker identity and recording information. In this figure the actual spectrum for the vowel pointed to by the mouse is also shown at the bottom of the figure. This allows the user to interact with the analyzed data.



F1: 4.25 Bark (419 Hz)
F2: 12.78 Bark (1986 Hz)
Phoneme: /i/
Transcription: s/u/b-/s/k/r/i/p-
In word: "subskriptio"
Speaker: Martti Vainio

Figure 6. Interactive F1/F2 chart, information related to a specific /i/ vowel (pointed to by the mouse), and its corresponding auditory spectrum.

SUMMARY

This paper presented a powerful and flexible object-oriented speech database system. The tools used to transcribe signals, the database system, as well as the phonetic hierarchy were described. Finally, some speech analysis methods were presented. Since the system is built on top of the extendible Lisp, CLOS, and QuickSig DSP substrate, users are free to add new analysis methods according to their needs.

REFERENCES

[1] Esprit Project 2589 (SAM) Final Report.

[2] Karjalainen, M. and Altosaar, T. (1993) An Object-Oriented Database for Speech Processing. EuroSpeech-93, Berlin.

[3] Karjalainen, M. (1990) "DSP Software Integration by Object-Oriented Programming: A Case Study of QuickSig." IEEE ASSP Magazine.