Finnish and Estonian Speech Applications Developed on an Object-Oriented Speech Processing and Database System

Toomas Altosaar[1], **Matti Karjalainen**[1], **Martti Vainio**[2], **Einar Meister**[3]

[1]Acoustics Lab., Helsinki University of Technology, Finland
[2]Department of Phonetics, University of Helsinki, Finland
[3]Laboratory of Phonetics and Speech Technology, Institute of Cybernetics, Estonia

Abstract

QuickSig, an object oriented signal processing system that represents a modern tool with which to perform DSP related studies, is presented. It empowers speech scientists to operate in a flexible and motivating environment where signals, filters, spectrograms, etc., are all modelled as objects. Seamlessly integrated to QuickSig is an object-oriented database that permits signals along with their features and relations to be stored persistently between sessions in a manner that is transparent to the user. A multilingual phonetic representation system exists within the same environment and allows speech from different databases to be modelled generically. Relations between speech units such as sentences, words, phones, etc., are defined explicitly forming a phonetic object structure for each utterance. Complex pattern matching searches can be easily formulated by the user and made to traverse the phonetic structures returning desired contexts. These speech events can then be used in actual applications that have been developed on this platform. Finnish and Estonian databases have been used as the source speech material for applications that include speech synthesis, recognition, and speaker verification/identification.

1. Introduction

Comprehensive utilisation of information available in speech databases has not always been feasible due to the differing standards and formats employed. In addition, the extra diversity introduced by the multilingual aspect has made the analysis of speech databases even more difficult under a single computing environment.

After technology enabled the collection and storage of large amounts of digitised data it was realised that speech databases would represent an indispensable resource for spoken language studies. Speech databases have been produced for several of the world's languages and more are currently being designed. No definite standard exists for the format of databases but usually a typical collection contains sentences or words represented by a set of files. Each utterance file may also have one or more files containing an orthographic transcription or transliteration, a phonemic transcription, phonetic labels, and segment boundaries. Additional information related to the recordings may be, e.g., speaker identity, dialect, and recording environment. Language specific elements, e.g., special character sets, phonetic alphabets, orthography, must be taken into account if the full potential of the information supplied is to be of use.

The emergence of a common standard for representing speech seems even more unlikely when new databases are being created with a variety of related information with respect to the audio waveform. For example, including video for capturing facial movements and glottal waveforms for accurate pitch determination in terms of added value for future studies is indisputable. These additional features will also require new description standards.

This lack of a common standard makes the model that represents speech in a computational environment even more important. For information in speech databases to be put to full use, a flexible model must exist that allows the speech scientist to perform studies in an intuitive, logical, and motivating manner. To perform complex searches across databases of different languages, formats, and labels, a generic multilingual model of speech must be sought for that permits analyses in a common environment. Inefficient and cumbersome string searching operations on phonetic/phonemic labels should be replaced by a compiled structural object hierarchy — which captures the essence of the speech process — where searches are performed by advanced object matching predicate functions.

This paper reports on one such generic speech processing environment where the relationships between different speech objects such as phones, words, syllables, and sentences are explicitly represented. First an overview of the object-oriented computational environment, database, multilingual phonetic representation system, hierarchical structure, and database access infrastructure is given. Specific applications can benefit from this generic speech modelling system and examples of synthesis, recognition, and verification/identification systems developed for Finnish and Estonian are then given.

2. Object-Oriented Environment

2.1 QuickSig

The system is based on QuickSig (Karjalainen & Altosaar & Alku, 1988), an experimental DSP programming environment that is application independent. Being implemented in Common Lisp with CLOS (Common Lisp Object System) it can be seen as a signal processing extension to the Lisp language. QuickSig retains the elegant syntax of the Lisp language, i.e., `(function arg1, arg2, ...)`, to the domain of signal processing by forms such as `(add sig1 sig2)` which would add two signal objects sample by sample and create a new signal object as the return value of the function *add*. QuickSig consists of object classes and method functions that enable the definition of complex signal processing algorithms via the class inheritance mechanism of CLOS. Filters, analysers, and signal displays such as spectrograms are all implemented as classes and instances leading to an environment where speech studies can be flexibly carried out. QuickSig is extendible by the user and new algorithms can be defined and tested immediately due to incremental compilation. Figure 1 shows the Lisp substrate onto which QuickSig has been fused. Other higher layers are described in the following subsections.

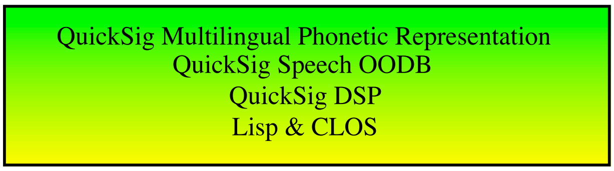

Figure 1: Layers in the QuickSig system: the DSP extension to the Lisp language, the object-oriented speech database, and the multilingual phonetic representation system all seamlessly integrated.

2.2 Object-Oriented Database

TIMIT, EUROM and Kiel are examples of speech databases with a fixed and limited number of parameters, e.g., waveform, transcription, segments, etc. For a database to be of use to the user additional features are frequently required to be calculated, e.g., long-term spectra, loudness curves, etc. It would be convenient to be able to have access to these additional features immediately when required from an object cache that is transparent to the user. Furthermore, if the additional data cannot be represented as simple records of data items but requires complex relations and data structures for its representation then a fixed format approach often becomes strained and susceptible to errors. In many speech research tasks a highly flexible way of doing inquiry operations in a large and complex database, e.g., by rule expressions, has become even more important.

The object-oriented database (OODB) existing in QuickSig is based on object-oriented programming (OOP) methodology. QuickSig allows for automatic persistent storage and updating of arbitrary data types and relations for the storage of speech signals, transcriptions, segments and related information in a flexible way (Karjalainen & Altosaar, 1993). Objects in non-permanent working memory are backed up to permanent storage and after restarting the database system of objects needed for computation are loaded back to working memory only when required. The file system remains hidden to the user since bookkeeping of data in permanent and working memory is automatic and provided by the system.

Files existing in the database have been for efficiency reasons divided into two basic types: a) files containing Lisp forms, when evaluated, recreate any type of object and its relations that existed prior to being placed in permanent storage, and b) specialised binary files that represent signal samples efficiently and permit their fast transport between working and permanent memory. AIFF (Audio Interchange File Format) type of files have been used for the latter since their standard permits the definition of data chunks which can be used to define and store arbitrary data structures.

Existing databases can be loaded, have new computationally expensive features calculated and linked to their elements, and then be dumped to permanent storage in a more flexible form. This allows for these new features to be efficiently retrieved via the transparent links. As seen in figure 2, databases can be viewed and actively modified using the database grapher.

2.3 Multilingual Phonetic Representation

The lack of universal standards in speech databases regarding labeling symbols and to a lesser extent audio data has made the development of multilingual speech database access systems difficult. A common formalism for representing the sounds of any of the world's languages by Worldbet (Hieronymus, 1993) addresses the former of these difficulties.

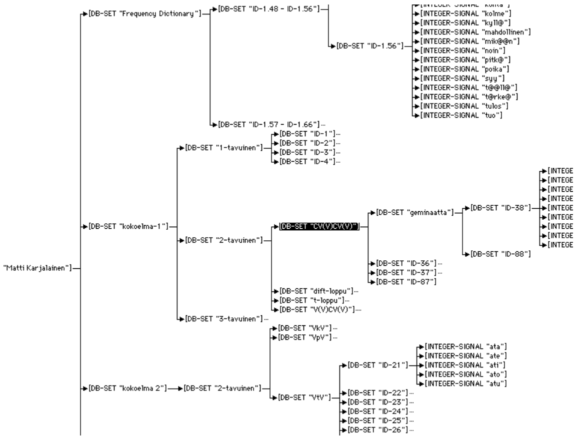

Figure 2: Part of the Finnish speech database as viewed with the database grapher.

Worldbet is an ASCII version of the International Phonetic Alphabet (IPA) and has in addition broad phonetic symbols not currently in IPA. Separate symbols for each different speech sound along with diacritics to describe allophonic variation cover the entire repertoire of human vocalisation. Existing phonetic alphabets used for labeling speech corpora such as TIMITBET and SAMPA can be mapped onto Worldbet. Since Worldbet provides a common description of a speech sound, a phone labeled in one phonetic alphabet can be mapped to its corresponding symbol in another alphabet. This may be of use for a phonetician studying another speech database that has been labeled in an unfamiliar alphabet. Figure 3 shows the relationship between Worldbet and other existing phonetic alphabets.

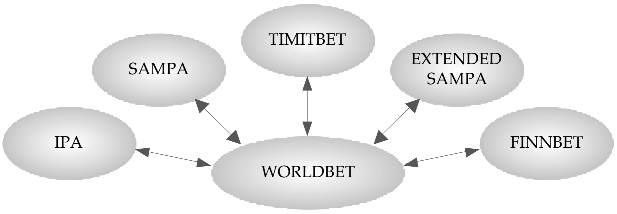

Figure 3: Relationship between Worldbet and other phonetic alphabets.

In the QuickSig phonetics package Worldbet symbols are seen as classes that can be decomposed into their constituent phonetic features (Altosaar & Karjalainen & Vainio, 1996). A Worldbet class is defined by combining lower level feature classes, e.g., the class /E/ (/ε/ in IPA) combines the features *open-mid*, *front*, *unrounded*, *monophthong-vowel* as well as *voiced*. Figure 4 shows part of the phonetic class hierarchy used to define the Worldbet class /E/.

A separate parser is required for each different format of database to decode and compile the phonetic/phonemic labels, orthographic/transliteration text, and segment locations from string format into instances of the defined classes.

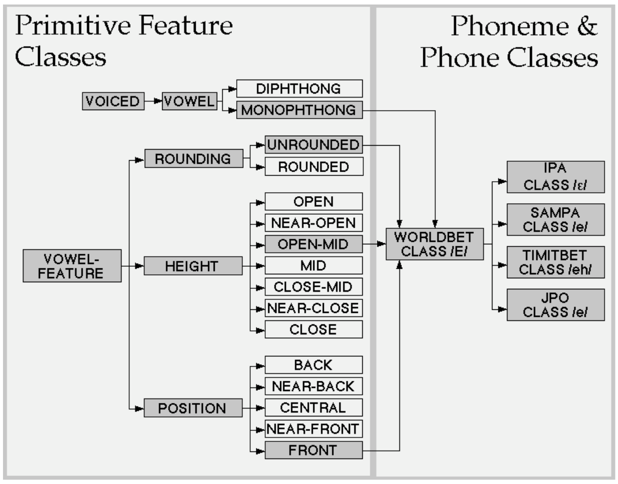

Figure 4: Class hierarchy used to define the Worldbet class /E/ as well as other derived phonetic alphabets.

For example, figure 5 shows how data representing a phone from both the Kiel and TIMIT database are converted into a phone object that encapsulates knowledge about its constituent features via the class hierarchy. Speech objects such as phones are linked to the originating signal object and contain knowledge of their span within the waveform.

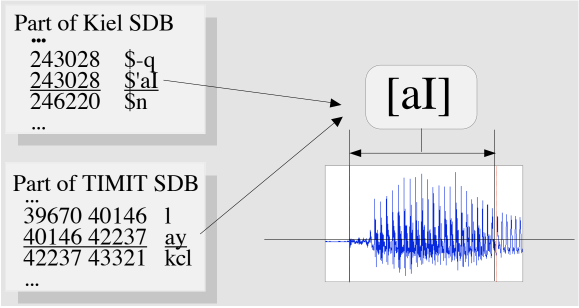

Figure 5: Parsers convert textual representations of speech into actual speech objects that contain knowledge about their feature makeup which is exploited during the database access phase.

2.4 Hierarchical Phonetic Structure

Other structural classes such as sentence, word, syllable, segment, etc., are used to define a set of discrete levels across the different temporal scales of speech. All instances of these classes can incorporate substructure, i.e., a sentence can point to a list of words and each word can have in turn a set of phones. All these different layers are linked together to form a hierarchical structure of speech objects that facilitates database searches. Figure 6 shows the phonetic structure that results from parsing an utterance from the TIMIT database.

Each of the speech units knows its relative position in this hierarchy in both a top-down manner, e.g., from sentence level down to segment level, as well as in a left-right, i.e., temporal order. All the objects are bidirectionally linked forming a network where traversal is possible. To represent a pair of primary units a set of di-units is used. This is indicated in figure 6 for the phone level by an additional diphone level.

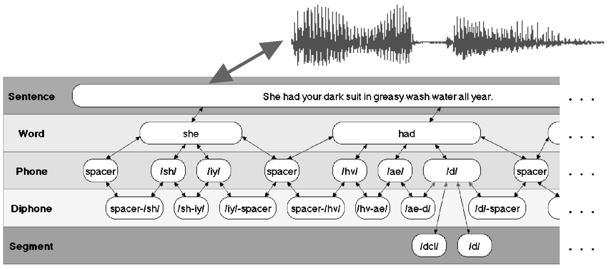

Figure 6: Part of the hierarchical structure representation for the TIMIT sentence: “She had your dark suit in greasy wash water all year”. Ovals and circles represent instances of phonetic classes while double ended arrows indicate bidirectional links. Some links, e.g., from phone to adjacent phone, are computational for memory efficiency reasons and are not shown.

2.5 Database Access

Database searches have been typically performed using a rigid string matching paradigm based on a textual representation of speech. The availability of a compiled and rich hierarchical phonetic structure allows predicate functions to traverse the networks and identify desired contexts.

Utilising the comprehensive class hierarchy that defines Worldbet and its derived phonetic classes enables searches to be carried using the efficient class inheritance mechanism that exists within Lisp and CLOS. A library of low level functions, relations, and types is used to construct predicate functions. Functions such as `next-phone` can be applied to any phone object to gain access to the following phone, even across word or sentence boundaries. In general, relation functions such as `prev`, `next`, `part-of`, and `parts` can be applied to any speech object to move around in the network. Speech objects can be tested for class membership by primitive Lisp functions such as `typep` and `type-of`, e.g., `(typep x 'open-mid)` would return a non-nil value for any arbitrary phone x if it inherited the *height* feature class *open-mid*.

Search predicates can be formulated in Worldbet or any other defined phonetic alphabet due to the class inheritance hierarchy. For example, a search predicate can be defined using the SAMPA phonetic alphabet and applied to a database labelled with TIMITBET, assuming that the required phones exist within both alphabets.

The following example defines a predicate function that can be used to search for [m] phones surrounded by vowels spoken by male speakers:

```
(defmethod VmV ((x phone))
  (and (typep x (Worldbet "m"))
       (typep (prev-phone x) 'vowel)
       (typep (next-phone x) 'vowel)
       (eq (gender (speaker x)) 'male)))
```

In this case a generic function called VmV is defined by *defmethod* that is specialised to operate on any phone object represented by the variable x. The phone to which this predicate function is applied requires that all four sub-predicates return a non-nil value for the function itself to return a non-nil value: x is first tested for class membership of the Worldbet class `/m/`, then its left and right neighbours are tested that they inherit the class `vowel`, and finally that the speaker is of the gender `male`. Phones as well as all other speech objects can determine the signal they represent via links and likewise signals are linked to their respective speaker objects where age, gender, dialect and other properties are stored. If all of the conditions return a non-nil value then the entire function returns a non-nil value indicating a match, otherwise the function terminates immediately if any nil value is encountered. User defined predicate libraries can easily be defined and developed since search predicates can also incorporate any defined function in the environment.

A search returns a list of objects that match the predicate function. These may be, e.g., phones, words, segments, or in general any object. These objects can be operated upon immediately by a wide variety of signal processing operations available in the QuickSig environment. Searches can be applied to an entire database or to selective parts of it, or to several different multilingual databases concurrently. A common graphical user interface exists for all databases which permits the user to control the search space effectively.

As an example of database access the VmV predicate defined above is applied to two speech databases containing recorded speech of different languages: TIMIT (American English) and a collection of Finnish speech. Applying it to the TIMIT database yields 518 contextual matches while only 162 are found in the Finnish database. The matches are returned as a list of the actual phone objects existing in the hierarchical phonetic structure with all links intact. By calculating the average auditory spectrum and distribution of each set of phonemes figures 7 and 8 are produced, for TIMIT and the Finnish database, respectively. By defining a new predicate that searches for VnV contexts instead — by simply changing the "m" to "n" in the above VmV predicate and renaming the function to VnV — and applying it to the same databases with similar post signal processing yields figures 9 and 10. Axes are loudness (relative sone) vs. pitch (Bark).

3. Applications

In this section some of the applications that have been developed using the above described system are briefly covered. In all cases either Finnish or Estonian speech databases have constituted the language resources required to generate the training and evaluation material for each specific application. Hierarchical phonetic representations have also been used in each of the following applications to provide easy access to the speech data. The speech data itself was used to generate training and evaluation material for neural networks all of which were of the multilayer feed-forward type taught using a standard backpropagation algorithm. All these procedures and objects existed in the QuickSig system seamlessly.

3.1 Speech Synthesis

A prototype for improved text-to-speech (TTS) synthesis for the Finnish language was developed (Karjalainen & Altosaar & Vainio, 1998) using the object-oriented speech database formalism presented in section 2. Warped linear prediction (WLP) was used as a speech production model yielding wide audio bandwidth while enabling compact control parameter representation. Figure 11 illustrates the synthesiser in a block diagram format.

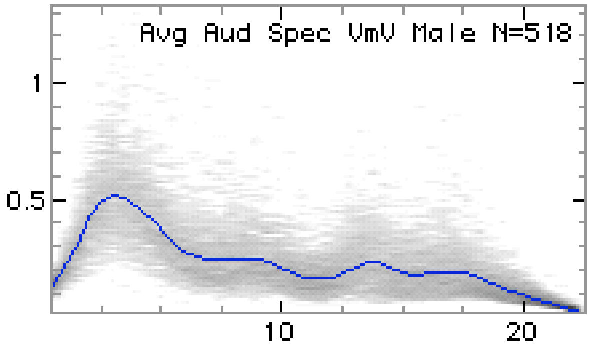

Figure 7: Average auditory spectra and distribution for [m] in a VmV context. Male TIMIT speakers only.

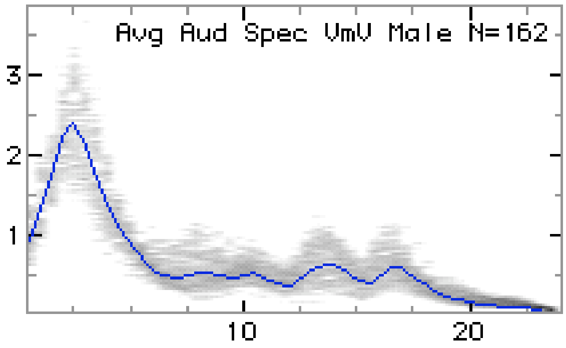

Figure 8: Average auditory spectra and distribution for [m] in a VmV context. Male Finnish speakers only.

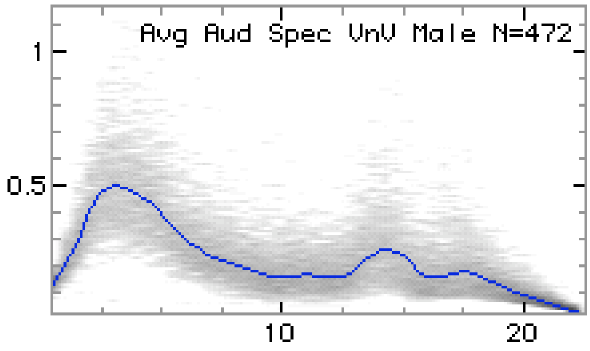

Figure 9: Average auditory spectra and distribution for [n] in a VnV context. Male TIMIT speakers only.

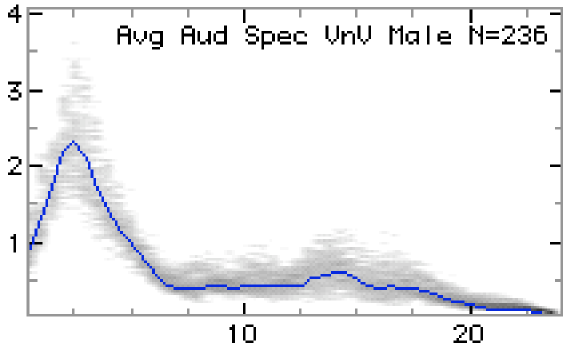

Figure 10: Average auditory spectra and distribution for [n] in a VnV context. Male Finnish speakers only.

The structure consists of an excitation codebook, an overlap-add concatenator of excitation signals for pitch and duration generation (Vainio et al. 1997), a gain multiplier, and a WLP filter. Sets of context-specialised neural networks called *netsets* control the synthesis chain's filter parameters, pitch, duration, and gain. Neural network inputs as well as the selection of a proper network within a netset is based on the phoneme to be synthesised, its phonemic context as well as other information available from the context.

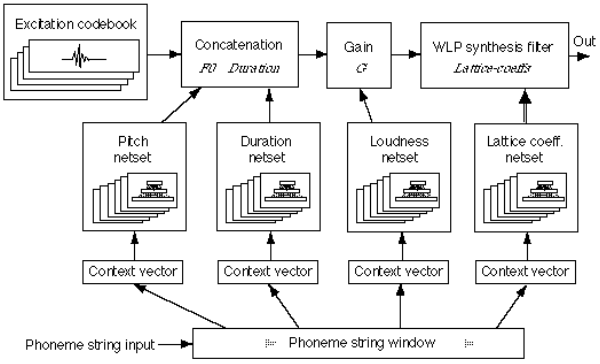

Figure 11: Speech synthesis using warped linear prediction controlled through sets of neural networks that have been trained on data from a Finnish speech object-oriented database.

The detailed control parameters for synthesis are generated using a set of specialised neural networks that react to phonemic input in specific contexts. Each network contributes only within a specific region of a multidimensional data space. Such a configuration has proven to function better than one single large network.

Over 2000 words spoken in isolation by a single male speaker constituted the database. Each signal had been manually segmented and labelled and a hierarchical phonetic structure calculated for it. The words were split into two sets with a 2:1 ratio, one for training and the other for evaluation, respectively. Different predicate functions were designed to access selectively training and evaluation material from the database.

Predicates were applied and in excess of 100,000 training and 50,000 evaluation elements were found within a few seconds and applied to neural network training and evaluation immediately. This all occurred in the same computational environment where the user could inspect any object in memory, or, decide what to do next without having to switch applications or compile and link new software.

Figure 12 shows the performance of the coefficient filter for the transition [e]-[i] in the Finnish word /keinu/. Here the actual target auditory spectrum is displayed as the top-most curve. The four following curves are neural net generated spectra in order of decreasing network specialisation: /e/–/i/, front-vowel–front-vowel, vowel–vowel, and *–* where * represents any phone. The network with the highest degree of specialisation produced the spectrum (second top-most curve) that in shape was closest to the target (top curve).

3.2 Speech Recognition

In a similar manner to the speech synthesis experiments described in section 3.1, experiments have been carried out using the same database material but for use in a diphone-based speech recogniser (Altosaar & Karjalainen, 1992). Diphones are used as the primary recognition unit in a time-event neural network (TENN) framework where recognition is carried out in a two-step approach: coarse event detection followed by fine level classification.

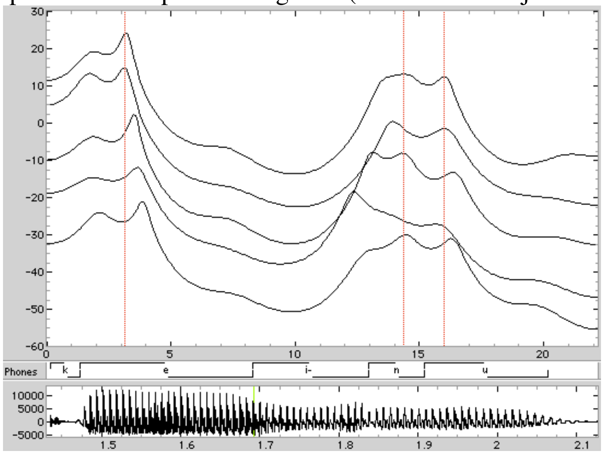

Figure 12: Warped linear prediction spectra within the [e]-[i] transition of the Finnish word /keinu/ (dB vs. Bark scale). The topmost curve is the target spectrum and the other ones are neural network generated cases in order of decreasing specialisation.

In the first step a diphone is localised in time. This is accomplished by a set of specialised neural nets that have been trained to detect coarse-class phone-phone transitions such as vowel–stop, nasal–vowel, etc. Seven different coarse classes were trained which cover approximately 70% of normal Finnish. The networks were taught at 5 ms intervals to generate an output value of 0.1 when they were not in the ±10 ms vicinity of a phone–phone transition. Within a transition the target value for a network was assigned the value 0.9.

By presenting temporally shifted versions of diphones to the trained nets over a ±60 ms interval the separate responses could be summed together to better study the temporal selectivity. Figure 13 shows normalised analogue responses for the stop–vowel and vowel–stop coarse class diphone detectors. As can be seen the nets exhibit good selectivity around the desired diphone.

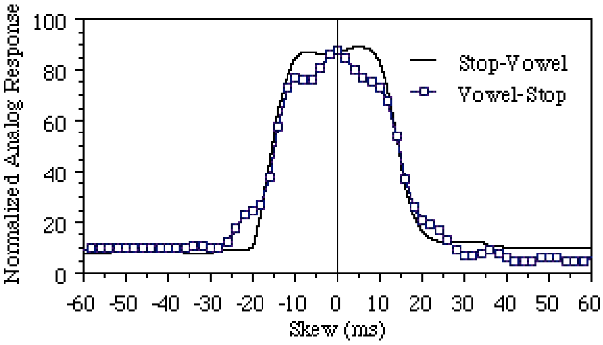

Figure 13: Temporal diphone response of stop-vowel and vowel-stop detector nets: dimensions of the neural nets were 168–20–1 and 120–20–1 nodes, respectively.

3.3 Speaker Recognition

The QuickSig platform has also been used to perform research on speaker recognition (Altosaar & Meister, 1995). These experiments were carried out on an Estonian speech database containing 43 sentences from 5 male and 5 female speakers each. In addition to this material one minute of speech from 10 extra speakers (5 males, 5 females) was also recorded to act as impostor material. Therefore, the experiments were based on 540 utterances from 20 native Estonian speakers — about 40 minutes in total.

A new database object was first created, then speech was recorded in an office environment and added to the database, and finally a set of spectral features calculated and linked to the signals. Objects such as speakers were also linked to the signal objects so that training material generation would be facilitated.

Figure 14 illustrates the general strategy and hierarchy of the proposed system. Speech is first classified as either noise, male or female. Spectral and temporal features are extracted and passed to feature recognisers that evaluate the evidence.

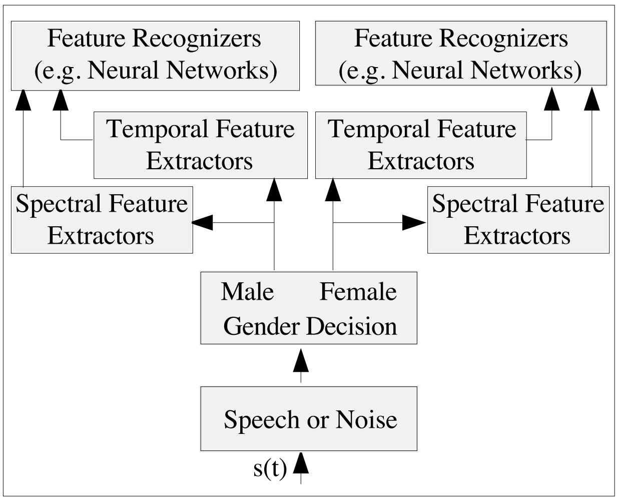

Figure 14: Hierarchy of the speaker recognition system.

In these experiments studies were performed to find reliable spectral features as well as good network topologies for their recognition. Three different spectral representations were studied: loudness (linear frequency scale), auditory (Bark scale), and warped linear prediction. Both long and short term spectra were used as training material. Four different network topologies were studied ranging from a general recogniser, gender specific networks trained on only one sex, and networks that performed speaker verification. Figure 15 shows the different types of configurations used for the networks.

It was found that long-term spectra outperformed short-term spectra and that loudness and auditory spectra performed better than a compact WLP representation. Specialisation of networks again proved useful since networks where only a single speaker was taught (NT4) performed well.

4. Discussion

Lisp and CLOS based QuickSig has proven to be a flexible, efficient, and robust tool to perform studies on speech. Explicitly modelling the speech process using the OOP paradigm has aided speech scientists to perform experiments in a motivating and advanced environment.

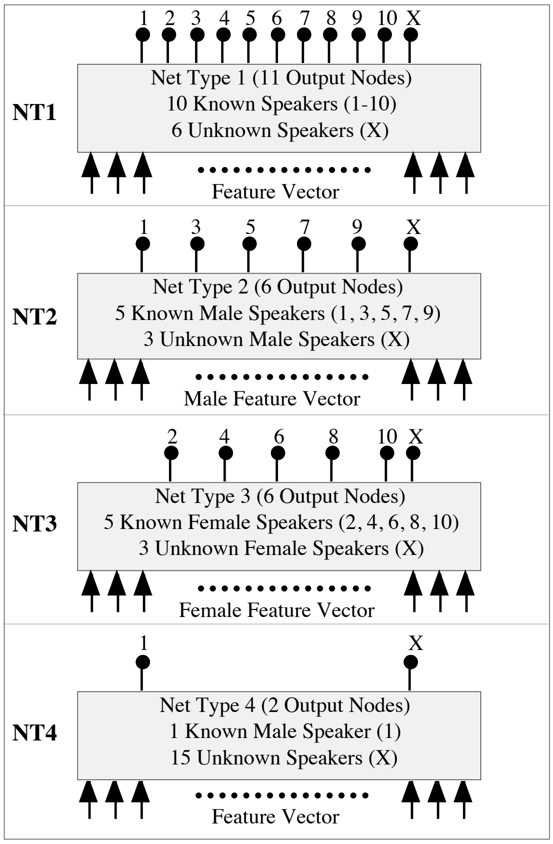
1 2 3 4 5 6 7 8 9 10 X
NT1
Net Type 1 (11 Output Nodes)
10 Known Speakers (1-10)
6 Unknown Speakers (X)
Feature Vector
1 3 5 7 9 X
NT2
Net Type 2 (6 Output Nodes)
5 Known Male Speakers (1, 3, 5, 7, 9)
3 Unknown Male Speakers (X)
Male Feature Vector
2 4 6 8 10 X
NT3
Net Type 3 (6 Output Nodes)
5 Known Female Speakers (2, 4, 6, 8, 10)
3 Unknown Female Speakers (X)
Female Feature Vector
1 X
NT4
Net Type 4 (2 Output Nodes)
1 Known Male Speaker (1)
15 Unknown Speakers (X)
Feature Vector

Figure 15: Speaker recognition (NT1) to identification (NT4) neural network topologies used in the experiments.

References

Altosaar T., Karjalainen M. (1992). Diphone-Based Speech Recognition using Time-Event Neural Networks. In Proceedings of ICSLP-92, Banff, Canada 1992.

Altosaar T., Karjalainen M., Vainio M. (1996). A Multilingual Phonetic Representation and Analysis System for Different Speech Databases. In Proceedings of ICSLP 96, Philadelphia.

Altosaar T., Meister E., (1995). Speaker Recognition Experiments in Estonian using Multi-Layer Feedforward Neural Nets. In Proceedings of EUROSPEECH-95, Madrid.

Hieronymus, James. L. (1993). ASCII Phonetic Symbols for the World's Languages: Worldbet, Bell Labs Technical Memorandum.

Karjalainen M., Altosaar T., Alku P. (1988). QuickSig - An Object-Oriented Signal Processing Environment. In Proceedings of IEEE ICASSP-88, New York.

Karjalainen M., Altosaar T. (1993). An Object-Oriented Database for Speech Processing. In Proceedings of EUROSPEECH-93, Berlin.

Karjalainen M., Altosaar T., Vainio, M. (1998). Speech Synthesis using Warped Linear Prediction and Neural Networks. In Proceedings of IEEE ICASSP-98, Seattle.

Vainio M., Aulanko R., Altosaar T., Karjalainen M. (1997). Modeling Finnish Microprosody for Speech Synthesis. ESCA Workshop on Intonation Theory and Applications (pp. 309--312). Athens.