HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Laboratory of Acoustics and Audio Signal Processing

Okko Räsänen

Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, November 5, 2007

Supervisor: Professor Unto K. Laine
Instructor: Professor Unto K. Laine

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Okko Räsänen
Name of the thesis:	Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture
Date:	November 5, 2007 **Number of pages:** 77 + 17
Department:	Electrical and Communications Engineering
Professorship:	S-89
Supervisor:	Prof. Unto K. Laine
Instructor:	Prof. Unto K. Laine

To reduce the gap between performance of traditional speech recognition systems and human speech recognition skills, a new architecture is required. A system that is capable of incremental learning offers one such solution to this problem.

This thesis introduces a bottom-up approach for such a speech processing system, consisting of a novel blind speech segmentation algorithm, a segmental feature extraction methodology, and data classification by incremental clustering. All methods were evaluated by extensive experiments with a broad range of test material and the evaluation methodology was itself also scrutinized. The segmentation algorithm achieved above standard quality results compared to what is found in current literature regarding blind segmentation. Possibilities for follow-up research of memory structures and intelligent top-down feedback in speech processing are also outlined.

Keywords: speech segmentation, speech clustering, data classification, feature extraction, speech perception, pattern recognition, bottom-up processing, top-down processing

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Okko Räsänen
Työn nimi:	Puheen segmentointi ja klusterointi uutta puheentunnistimen arkkitehtuuria varten
Päivämäärä:	5.11.2007 **Sivuja:** 77 + 17
Osasto:	Sähkö- ja tietoliikennetekniikka
Professuuri:	S-89
Työn valvoja:	Prof. Unto K. Laine
Työn ohjaaja:	Prof. Unto K. Laine

Perinteiset automaattiset puheentunnistusmenetelmät eivät pärjää suorituskyvyssä ihmisen puheenhavaintokyvylle. Voidaksemme kuroa tämän eron umpeen, on kehitettävä täysin uudentyyppisiä arkkitehtuureja puheentunnistusta varten. Puhetta ja kieltä itsestään ihmisen lailla oppiva järjestelmä on yksi tällainen vaihtoehto.

Tämä diplomityö esittelee erään lähtökohdan oppivalle järjestelmälle, koostuen uudenlaisesta sokeasta puheen segmentointialgoritmista, segmenttien piirteistyksestä, sekä menetelmistä vähittäiselle puhedatan luokittelulle klusteroinnin avulla. Kaikki metodit arvioitiin kattavilla kokeilla, ja itse arviontimenetelmien luonteeseen kiinnitettiin huomiota. Segmentoinnissa saavutettiin alan kirjallisuuteen nähden hyvät tulokset. Järjestelmän mahdollisia jatkokehityssuuntauksia on hahmoteltu muunmuassa mahdollisten muistiarkkitehtuurien ja älykkään top-down palautteen osalta.

Avainsanat: puheen segmentointi, puheen klusterointi, äänimateriaalin luokittelu, piirteistys, hahmontunnistus, puheen havaitseminen, bottom-up prosessointi, top-down prosessointi

Acknowledgements

All research was conducted in the Laboratory of Acoustics and Audio Signal Processing at the Helsinki University of the Technology. This thesis is part of the work in the Acquisition of Communication and Recognition Skills (ACORNS) project, funded by the Future and Emerging Technologies, in the Information Society Technologies thematic priority in the 6th Framework Programme of the European Union.

I would like to thank my supervisor and instructor Unto K. Laine for the invaluable ideas and guidance that have made this work possible in the first place. I would also like to thank Toomas Altosaar for all of the help and useful discussions during this work period. Finally, a big thanks goes to Martti Rahkila for his aid in all practical arrangements related to my work and studies in the laboratory.

Espoo, November 5, 2007

Okko Räsänen

Contents

List of abbreviations

A1	Primary Auditory Cortex
APA	Automatic Phonetic Annotation
ASR	Automatic Speech Recognition
ASS	Automatic Speech Segmentation
CV	Consonant-Vowel [pair]
DFT	Discrete Fourier Transform
F1	First Formant
F2	Second Formant
FFT	Fast Fourier Transform
FE	Feature Extraction
HMM	Hidden Markov Model
LP	Linear Prediction
LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficients
MNS	Mirror Neuron System
MSE	Mean Squared Error
OAE	Otoacoustic Emission
PLP	Perceptual Linear Prediction
PM	Primary Memory
RA	Reference Annotation
RASTA-PLP	Relative Spectral Perceptual Linear Prediction
SM	Secondary Memory
STE	Short-Term Energy
ZCR	Zero Crossing Rate

1 Introduction

Information sharing is one of the most salient features of the human race. The ability to communicate is like glue, binding groups of people together and making social organizations possible. Without a proper way of communication, information sharing would be very limited and it would not extend far from the practical physical aspects of our daily lives. As a result of evolution, our species has developed a very special mechanism for communication: language. In the light of current scientific knowledge, the development of language has in parallel evolved with our speech organs, both driving each other towards an optimal way to transfer information over a challenging medium, namely air (*Fitch*, 2000). Language also aids our thinking conceptually, the way we perceive and interpret the world around us (*Gauker*, 2002). Even more significantly, without language we could not perceive ourselves in the same way as we are doing at this very moment.

But how do we actually learn language as infants? Do our brains have some sort of built-in system for processing linguistic information? Current knowledge of the cognitive aspects of speech processing is controversial. There is strong evidence that specific brain areas are specialized in speech processing (*Boatman,* 2003; *Stefanatos et al.*, 2007; *Burton et al.*, 2000), but the same areas are also known to be involved in many other processes (*Gazzaniga*, 2002). At this moment there is no certainty if these areas are specialized for language, or, if their processing abilities are just exploited by language. Recently, *Price et al*. have argued that brain regions activated by speech processing are only harnessed by differential demands on auditory processing, and could be re-classified to respond to other types of stimuli (*Price et al.*, 2005). Interestingly, this supports old theories of generality regarding the processing principles in the neo-cortex (*Mountcastle & Vernon*, 1978) that have been lately revised in popular science literature by *Jeff Hawkins* (2004).

What is significant is that we know that the brain's and its billions of neural connections obey the laws of physics and can be described by some sort of statistics. If speech comprehension and learning, which in practice are activation and adaptation of neurons for incoming auditory signals, can be described by such statistical models, then it should be also possible to simulate such processes by using technical computational models. On the other hand, developing models capable (or incapable) of imitating some aspects of cognitive processes can give us more insight to the real world attributes inside our heads, and can be used to reinforce or discard existing theories of cognitive science and

psychology. However, building a system that is "born" into reality without any prior knowledge, and that would learn to understand and communicate by interaction with its environment, has been so far possible only in the field of science fiction. Nonetheless, a continuous increase in computational capacity and relatively recent developments in many interesting mathematical, cognitive scientific and information technology applications during the last few decades has cumulated to a point where it is possible to start testing theories and ideas with practical simulations. Since language plays such an important role in our development to become members of society and the way we perceive the surrounding world, it is a natural choice to investigate this field further with computational simulations.

However, before being able to understand speech, both humans and machines must first be able to extract meaningful information from the mere continuous stream of pressure variations existing in the medium. This starts as a so-called "bottom-up" process, in which air pressure variations are transferred through our outer and middle ear to the cochlea, where pressure variations then propagate in a liquid medium. However, already inside the cochlea the purely bottom-up process ceases and interaction between the input and feedback from the central neural system begins: inner hair cells in the basilar membrane respond to the propagating sound wave and send signals to the acoustic nerve leading to the auditory nuclei in the central neural system. Activation of these low-level processing "centers" initiates an immediate neural feedback, causing outer hair cells in the basilar membrane to adjust their stiffness according to the properties of the input, and results in a better discrimination capability for the incoming signal (*Ulfendahl & Flock*, 1998). It seems evident that neural feedback mechanisms start to play an ever increasing role as the processing progresses towards higher-levels of cognitive functions. This interaction between bottom-up and top-down signaling may contain the clue to the mystery that surrounds our cognition as a whole.

From the computational point of view, this bottom-up top-down interaction imposes an interesting challenge: if we want to build a system that can learn from speech, mechanisms for converting acoustic signals to language-like patterns are required. On the other hand, we need a system that knows something about the language and provides top-down information, before we can do human-like processing (e.g., correct word segmentation and recognition in extremely noisy conditions). This means that instead of just learning the language from bits of information supplied from below, our system also has to learn the process of *processing* itself.

While there is a plethora of knowledge available about the functionality of the human auditory system (e.g., *Moore*, 1995; *Karjalainen*, 1999), there is still much to be solved and understood, especially at the higher levels of processing. This leads to the question: how do we actually model language acquisition? How do we know what models to use to create some sort of memory-structure that can mediate between inputs and outputs? How do we know how speech should be organized in the system and what sort of feedback should be used at different levels? The answer is that we don't. But in computational modeling and simulations we can always exploit what we already know about our hearing and speech processing, or we can test new methods of processing that could

support empirical evidence. The technological advances of our society allow us to cling to fundamental mechanisms of cognitive processing by using computational methods.

To build a framework for a new architecture of learning for speech recognition systems, we need a reliable foundation on which learning can begin. This master's thesis concentrates on understanding and developing methods for temporal segregation of continuous speech, also called segmentation, feature extraction of the segmented speech, and classification of the segmental data. The general perspective leans strongly to the cognitive aspects of human brain processing, trying to bind together some connections between traditional speech processing methods and cognitive science research. This work is intended to draft a methodological front-end basis for an interactive speech recognition system, which, in the near future, is able to utilize its past experience to enhance the quality of its pre-processing and ultimately to learn the language at hand. The experimental focus in the scope of this paper is on blind speech segmentation, and we also briefly probe into the vast world of feature extraction and data clustering to sketch some possible paths for future work.

2 Speech processing by humans and machines

In order to extract and distinguish bits of information from speech, we humans need to be able to distinguish and organize different parts of the signal both temporally and by its amplitude (waveform). The inner ear provides us with a biological mechanism for processing such information, where in technical solutions there are mathematical transformations (e.g., the Fourier transform) that are commonly used for transforming time-domain information into the frequency-domain. We also need to be able to divide the speech signal into meaningful and coherent parts, a process that is called *speech segmentation*. Segmented speech must be further processed in order to create something we can define as information.

Automated segmentation of speech signals has been under research for over 30 years. It is of much interest as an important pre-processing part in most speech processing systems that are intended to obtain some useful features carrying information in the auditory channel. It is a necessity for phonetic analysis of speech (*Mermelstein*, 1975), audio content classification (*Zhang & Kuo*, 1999) and many applications in the field of automatic speech recognition (ASR), including word recognition (*Antal*, 2004). Research in ASR aims to develop artificial systems that are able to understand[1] and/or act on incoming speech, and in some cases like the ACORNS[2] project, to also understand the mechanisms of human speech perception. While research in the field of speech recognition has been intense for decades, few real breakthrough success cases exist. The old challenges of artificial speech perception are still relevant and unsolved.

The general idea of segmentation can be described as dividing something continuous into discrete, non-overlapping entities (*Kvale*, 1993). In speech segmentation, the basic idea of segmentation is to divide a continuous speech signal into smaller parts, where each of these segments has phonetical or acoustical properties that distinguishes it from neighboring segments. Segments can also be thought of as patterns, each segment differing from total randomness in a coherent and (statistically) perceivable manner. The

[1] The concept of understanding is ambiguous and can be defined in many ways depending on the context and the goals of the process. Human understanding of speech is largely based on the knowledge of the lexicon and the grammar of the language, even though, e.g., machines are commonly thought to "understand" if they are able to match incoming speech signals with predefined actions associated with the content of the speech.

[2] Acquisition of Communication and Recognition Skills, http://www.acorns-project.org

size of a segment can vary due to the purpose of its use and the nature of the methodology by which segments are created. Segmentation can be performed, for example, at the *segment*, *phone*, *syllable*, *word*, *sentence* or *dialog turn* level. Segmentation can be also used to distinguish different types of audio signals from large amounts of audio data, often referred to as *audio classification*. This thesis concentrates solely on the processing of speech signals as the entire field of research and discussion in automatic audio processing and segmentation is much wider in scope.

Traditionally segmentation has been created manually by trained phoneticians who use their phonetic and linguistic knowledge with the help of some statistics, e.g., spectral and waveform information, to analyze and segment speech. High quality manual segmentation is an extremely slow and tedious task and therefore expensive to produce, but paired with the annotation of phonetic labels it fulfills the need for a required standard for segmentation quality evaluation (*Toledano et al.*, 2003). Manually produced annotation is often used to train algorithms for segmentation and labeling purposes. Automatic, context independent and real-time speech processing systems obviously are not able to utilize manual annotations (since none exist), so self-sufficient (or partly aided) and intelligent algorithms are needed in order to process speech into smaller units. Their performance is then usually compared to manual segmentation(s) and quality indices derived to evaluate an algorithm's performance (*Toledano et al.*, 2003; see also e.g. *Aversano et al.*, 2001 or *Sarkar & Sreenivas*, 2005).

In practice automatic speech recognition, including automatic speech segmentation (ASS), face many difficult problems: in natural speech there are no pauses between words and many phones or even parts of words are recognized by their context. Co-articulation often causes canonically expected phones to be modified or completely go missing since the speech organs (e.g., lips, tongue, etc.) are in continuous movement during the pronunciation of words. Loudness, pitch, duration and other possible elements of prosody such as voice quality all affect the physical attributes of the signal and carry additional linguistic and affective detail. Although all this is an ingenious and extremely effective way of transferring information between two or more living beings, it also complicates automatic recognition to a point where even today's state-of-the-art systems fall far behind in error rates compared to humans (*Boves et al.*, 2007).

In this chapter some perspectives regarding human speech processing will be discussed followed by a brief review of some of the most common approaches that have been applied to speech segmentation and feature extraction. Data classification by clustering will be also discussed. At the end of the chapter, the use of top-down information in speech segmentation and its current status in the area of speech recognition research is examined.

2.1 Speech processing, a human perspective

Human perception exploits prior learned and experienced information in a top-down way to make sense of what we hear (e.g., *Norris et al.*, 2003). Learning is one of the most important capabilities of our species that allows adaptation to environmental requirements and iteration towards more optimal problem solving. This imposes a question whether it is even possible to provide accurate and consistent models of cognitive processes that rely only on bottom-up organized processing. In speech processing, segmentation is one problem where attempts have been made to find a solution in both bottom-up and top-down ways, although none have produced generally acceptable and robust results.

But how do we humans actually segment speech? Or do we segment at all? Is there an automatic sub-cortical system for pre-processing speech sounds for further cortical analysis, or is everything processed in uniform manner all the way to the association areas and memory processes? How do we (or children) separate different phones from a continuous speech signal if we cannot even separate different syllables before we learn to read (*Morais et al.*, 1979; 1986)? On the other hand, in the light of current scientific knowledge, we cannot even say for sure what is the size or nature of the units our cortex uses to analyze speech. Or are there different parallel processing paths with different architectures for processing different aspects of speech? There are several questions remaining unanswered in this area, and hopefully future research in speech processing and cognitive sciences can shed some light. A short discussion of some of the aspects of human speech processing that are relevant to automatic speech processing systems now follows.

2.1.1 Speech discrimination capabilities

The ability to differentiate between two stimuli carrying different meanings is as important for humans as it is for machines trying to make sense of speech input. All natural languages have evolved in a way in which larger entities, sentences, are made up of smaller parts, words. Every word carries a set of meanings, often of a metaphorical nature, which can be combined with other meanings to produce an entity that carries meaningful information from the producer of the sentence to the receiver. Each word in turn consists of smaller units called syllables, each of which are made up of phonemes. In theory, using the entire set of phonemes of a given language, it is possible to produce all possible speech messages carrying linguistic information by combining them in the correct order. In order to differentiate meanings of similar messages, very subtle acoustical differences need to be detected. In some languages, even the differentiation of separate phones is not enough, since the message is encoded using prosodic features, e.g.,intonation, stress, quantity, etc.

Based on the assumption of hierarchical structure of natural languages, we can consider a language consisting of a series of small information chunks that can be combined for information transfer and split later for comprehension. Speech discrimination is then the

ability to *a)* identify these meaningful blocks in a speech stream, and *b)* to distinguish phonologically different blocks from one other. Speech segmentation is speech discrimination, although the term often has emphasis on temporal aspects of chunking continuous signals into smaller units. This division into blocks should not be taken too strictly however. Speech production is after all the continuous movement of speech organs, and the core of the message that is to be transferred is coded over entire utterances. The smaller the blocks that we dissect, the more significant is the effect of the context. Also, we do not know yet what the size of the most meaningful blocks are, or whether we use different parallel levels of structure for comprehension. But what can we actually distinguish, and do we first have to learn it?

Automatic phoneme discrimination at the auditory level is essential for speech comprehension, keeping in mind that phonemes are usually defined as units of speech that distinguish different meanings from each other (*Laver*, 1994). Our ability to understand very rapid speech with divided attention, paired with limitations in the capacity of the working memory (*Neath & Surprenant*, 2003), excludes the need and the possibility to use conscious effort to distinguish every single phone in continuous streams of speech. While it is necessary to differentiate realizations of phonemes (phones) in the speech stream, it is not something that all people capable with normal communication can explicitly perform when required. Research has shown that adults are able to distinguish phones at the beginning and at the end of words, but for example the ability to distinguish syllables in the middle of words requires literacy (*Morais et al.*, 1979; 1986). This suggests that while the process of phone discrimination is essential, it may not always be a conscious process.

There is also another interesting aspect of phone recognition, in which learning has a central role: infants are capable in distinguishing between phones of practically any language (see, e.g., *Blumstein et al.*, 1987), but as they mature, their discrimination abilities endure strongly only in the language(s) they continuously hear and learn (*Trehub*, 1976; *Werker & Tees*, 1984). Furthermore, there are several studies pointing out that while infants are capable of phone level discrimination, a tendency to process larger units of speech at a time might become dominant as the baby starts to learn language (see *Swingley*, 2004). On the other hand, children of age 6-8 need more acoustical cues for consonant-vowel (CV) pair discrimination than adults (*Elliott et al.*, 1986), which may be due to the advantage of experience with the language that adults have. A study carried out by *Stager & Werker* (1997) suggests that infants would prefer to use more phonetic detail in speech perception, but use more general features in word-learning as they start to map the auditory stream into conceptual meanings after the age of 8 months. This supports the idea that the brain of an infant gathers statistical cues of the environment (including speech) continuously, and as they grow older, the statistical representations start to enable higher level structuring of the incoming information. The ability to distinguish non-meaningful differences becomes pruned as the capacity is allocated to strengthen the processing of meaningful distinctions of the language.

Concluding all of this from the segmentation perspective, the ability to hear differences between phone sized changes in words, does not necessarily dictate that incoming speech

is processed in phone-size units for recognition. Recognition can be also thought of as a pattern matching operation, where synaptic connections in several neural pathways are activated in a unique manner depending on the acoustical properties of the incoming speech. Top-down attention and expectations can be thought of to modulate this activation spreading and guide activation propagation to cortical areas that are specialized in corresponding processing. Supporting evidence for role of attention can be found, e.g., from studies of *Toro*, *Sinnet* and *Soto-Faraco* (2005), who demonstrated necessity of attention for word segmentation from the auditory stream. It is interesting to discern that the attention mechanisms and primary memory codification and retrieval are considered to reside in the same brain areas, in the frontal cortex, that is also activated during speech processing (see next subsection).

2.1.2 Speech and the brain

There is also a strong possibility that infants use multimodal information and motoric development to learn to segment their mother tongue. Also, learning to structure other languages in later life may be aided by visual information that supports auditory input. The mutual necessity of context and speech has been long known to be a prerequisite for small children to learn new words (*Benedict*, 1979). This sort of multimodal learning, often referred as grounded learning, has also been researched lately in the area of artificial speech recognition with promising results (e.g., *Roy*, 2005). From the perspective of cognitive sciences, recently discovered mirror neuron system (MNS) has caused substantial discussion about the way humans perceive and imitate in order to learn. According to current knowledge, mirror neurons work as a neural substrate that binds together perception and action, in which learning by imitation can be thought of as a forming of links between these two (see e.g., *Rizzolatti & Craighero*, 2004 for a broad description of MNS).

The multimodality of speech perception is supported by various other studies. In one experiment it was determined that visual information of lip and jaw movement excites neurons at the pre-motoric cortex (*Nishitani & Hari*, 2002), at the very location of Broca's area that is associated with speech perception and production. However, perhaps the most salient proof of unconscious interaction between vision and hearing is the so called *McGurk effect*: presenting conflicting visual information of mouth movements at the same time with an auditory phone stream causes miscomprehension of the auditory signal. For example, visual *[ga]* and auditory *[ba]* leads to perception of *[da]* (*McGurk & MacDonald,* 1976). Preliminary experiments also show that computational speech segmentation can be performed effectively by using a relatively low frequency bandwidth, containing mainly of the 1st and 2nd formants that are most visible in facial expression of articulation. Also, the first two "visible" formants are known to be sufficient for fair vowel recognition (*Peterson & Barney*, 1952). Based on these findings it is possible to even hypothesize that visual information about facial actions may help infants to learn to segment auditory speech into meaningful units. Unfortunately, there is little research available that covers the area of speech segmentation and language learning in blind or otherwise visually confined infants.

The neural substrate behind speech processing is not something that one can simply point out. There has been rigorous discussion whether different steps or aspects of speech recognition are divided into spreading activation into several different areas, or whether there are only a few central locations for the recognition process. The idea of signal specific pathway processing, i.e. processing speech sounds in separate systems, can be justified more easily, if one considers how sub-cortical auditory processing already leads to tonotopic mappings at the primary auditory cortex (A1) (*Kosaki et al.*, 1997). Actual speech recognition and comprehension is believed to occur at the cortical level and probably in more specialized areas than A1 (*Wissinger et al.*, 2001), but also in separate areas than processing of, e.g., meaningful environmental sounds or background noise (see *Boatman*, 2004). Speech-specialized and focalized cortical substrates would not only make more efficient processing possible (*McNealy et al.*, 2006), but may also play a role in interaction with working memory and consciousness. Strengthening the importance of connection between memory and hearing, *Boatman* (2004) also showed in her lesion studies that the inferior frontal lobe plays a critical role in phonological processing. *Wilson et al.* (2004) confirmed this with fMRI experiments, showing that superior portion of the ventral pre-motor cortex activates during attended speech perception. These results strongly support the old motor theory of speech perception (*Mattingly & Lieberman, 1985*), as inferior frontal lobes are associated with motoric actions and learning by activation of mirror neurons (*Rizzolatti & Craighero*, 2004). *Locasto et al.* (2004) suggest that frontal areas are activated for extraction of acoustic information and keeping it in memory for decision. On the other hand, *Burton, Small and Blumstein (2000)* have postulated that frontal activation is only a product of processing taking place at the auditory cortex, and may play a role only with working memory demands for further processing. As one can see, it may be very difficult to say where general auditory processing ends and language specific recognition begins. The temporal lobe, including the substrate for primary auditory processing, seems to be tightly integrated with frontal lobes that are classically associated with attention, motoric planning and memory encoding and retrieval.

2.1.3 Conclusions of human perspective

The entire literature and research on speech and cognition is too broad to review here conclusively. The field and its knowledge is scattered as small pieces, and extensive approved theories that fuse together many aspects of cognitive functioning do not either exist or are extremely hard to prove experimentally. Language is one of these functions that needs highly developed cooperation from several different, classically separated, cortical processes, including auditory processing, episodic memory and associations. Current brain imaging techniques allow us to test well defined hypotheses with limited accuracy, limited also to very well controlled experimental settings with a small number of variables under inspection. These reliability requirements rule out large-scale exploration of interconnected neural substrates and interacting cognitive processes lying beyond them, aiming to find central cause and effect relationships. The limitations of the

available methods and the lack of extensive knowledge in cognitive processing leave many aspects of our speech processing still unknown.

Do we have a specialized memory for words and each of their different forms, or is there a system that converges a series of phones into coherent patterns that can be associated with something already seen or heard before? Or does the truth lie somewhere between? It remains to be seen what the size of speech units are that we use to understand our fellow beings. Hopefully, speech recognition models and simulations relying on speech segmentation can shed some light onto this mystery. If it were possible to build a robust system that is capable to recognize speech by using prescribed segmentation principles, then it may also be possible to learn something relevant about human speech processing.

2.2 Methodological approaches to segmentation

Automatic speech segmentation methods can be classified in many ways, but one very common classification is the division to blind and aided segmentation algorithms. These two different approaches will be discussed in more detail below. This classification can also be performed based on the parameters used to describe the original signal (see next section), often leading to a division between model-based (most relying on linear prediction, LP) and model-free methods (*Li & Gibson*, 1996). The principle idea of model-based methods is to fit incoming data into some existing, often polynomial based, models describing sections of speech to obtain a fairly small amount of parameters to describe the incoming signal. Model-free methods often make use of spectral and/or time-domain properties of the signal, mapping key values to some space using distance metrics and finding points of interest as a function of time.

A central difference between aided and blind methods is in how much the segmentation algorithm uses previously obtained data or external knowledge to process the expected speech. Some systems can learn and adapt statistically to signals they are being fed with, or they can be taught beforehand with varying techniques and emphasis. The main idea of learning is to find some statistically relevant information from the speech that can be exploited to enhance the quality of the segmentation. The most common statistical methods in ASR and ASS are Hidden Markov models (HMMs), where key features of the signal are used for pattern recognition and most probable acoustic sequence calculation (see *Knill & Young*, 1997 for an extensive description).

2.2.1 Blind segmentation

The term *blind* segmentation refers to methods where there is no pre-existing knowledge regarding linguistic properties, such as orthography or the full phonetic annotation, of the signal to be segmented. *Sharma & Mammone* (1996) have listed such applications where blind segmentation can or needs to be applied: speaker verification systems, speech recognition systems, language identification systems and speech corpus segmentation &

labeling. In practice, requirements of real-time processing and speaker or language independency can usually be fulfilled only with systems that are able to function without prior external knowledge. However, this does not exclude the possibility that the system uses some earlier obtained knowledge about features of the speech, which usually refers to some sort of machine learning (see, e.g., *Bishop*, 2007) and the ideology of top-down processing outlined in many sections of this paper.

Due to the lack of external or top-down information, the first phase of blind segmentation relies entirely on the acoustical features present in the signal. This bottom-up processing is usually built on a front-end parametrization of the speech signal, often using MFCC, LP-coefficients, or pure FFT spectrum (*SaiJayram et al.*, 2002). These parametrization methods are shortly described in section 2.3. Tracking the behavior of chosen parameters can lead to cues for possible segment boundary discovery. The preliminary results of this blind bottom-up process can be improved by using sophisticated data classifying techniques to provide "intelligent" feedback for boundary detection.

2.2.2 Aided segmentation

Aided segmentation algorithms use some sort of external linguistic knowledge of the speech stream to segment it into corresponding segments of the desired type. Usually this means using an orthographic or phonetic transcription as a parallel input with the speech, or training the algorithm in advance with such data. Using the phonetic annotation of the speech stream also enables automatic segment labeling, which can be, for example, used to help manual annotation of large speech corpora. Naturally, a phonetic annotation for an input stream is only available in a few situations since it takes plenty of time and money to produce accurate descriptions of speech by manual labor. Orthographic transcriptions are less complex to produce and therefore also less expensive to create for speech data, but differences in the orthography of different languages is a serious limitation, and orthography may not always describe the content of the acoustical signal faithfully.

However, there are also methods for creating automatic phonetic annotations (APA), for example, by using the orthographic transcription and the speech signal together (see, e.g., *Schiel*, 1999). The quality[3] of the annotation can vary, and such systems are usually language dependant and need to be trained beforehand. These methods are still useful as such for some applications and they can also facilitate manual phonetic annotation. The increasing quality of APA may also dominate the annotation of large speech corpora in the near future, as manual annotation is too expensive and time consuming (*Greenberg*, 2003).

[3] Defining annotation quality is not a trivial task. Automatic annotations are often compared to manually created reference annotations (RA), but manual annotations are also prone to inter- and intra-subject variations. Often many phoneticians need to work on the same material to obtain a desired degree of consistency in the annotation (*Cucchiarini & Strik*, 2003).

One of the most common methods in ASR for utilizing phonetic annotations is with HMM-systems. Using annotated phones as models, different representations of phone realizations can be used to determine the optimal parameters for each model. This is referred to as isolated HMM training (*Knill & Young*, 1997). HMM-based algorithms have dominated most speech recognition applications since the 1980's due to their so far superior performance in recognition and relatively small computational complexity (see, e.g., *Juang & Rabiner*, 2004, for the evolution of HMMs in the field of speech recognition).

In real time applications it is naturally impossible to have any external knowledge of the input in advance. One common usage for semi-automatic segmentation is for example raw segmentation of speech corpora. Algorithms can be taught with the speech material in the corpus to achieve decent segmentation quality. This preliminary segmentation is then manually corrected by phoneticians to obtain an accurate segmentation of the corpus. Pre-training segmentation algorithms for specified material, e.g., a specific language or limited set of words or sentences, is also effective and commonly used in lexically or functionally limited ASR applications.

2.3 Parametrization and feature extraction

The time domain waveform of a speech signal carries all of the auditory information. However, keeping in mind the difference between data and information, the waveform can mostly only be considered as pure data. From the phonological point of view, very little can be said on the basis of the waveform itself. However, past research in mathematics, acoustics, and speech technology have provided many methods for converting data into something that can be considered as information if interpreted correctly. A large proportion of speech technology research concentrates on the processing of auditory signals in such a manner that essential information of the conveyed message can be extracted from ambiguous data, which is potentially corrupted by (environmental) noise and the distortional properties of the transfer medium. As a result of these processing operations, a number of parameters or features are created.

Features can be used to depict changes in the signal as a function of time in a compact form and often with relatively low computational costs, which may be suitable as a basis for locating possible word or phone boundaries. Besides, in many ASR systems segmentation is only an inevitable preliminary operation for many methods. In order to find some statistically relevant information from incoming data, it is important to have mechanisms for reducing the information of each segment in the audio signal into a relatively small number of parameters, or features. These features should describe each segment in such a characteristic way that other similar segments can be grouped together by comparing their features, but differing segments (in terms of phonetic content) will be excluded. In an optimal situation, all sub-word segments that carry the same linguistic information would form a single group. In practice, this classification is extremely difficult to achieve, considering that the human auditory system is the most optimized

speech recognizer known so far, and even it produces classification errors at low levels of noise, especially for small units as phones.

For feature extraction methods it seems sensible to reduce the amount of acoustic information by using auditory models that try to imitate the functionality of human hearing. However, it is still controversial whether speech recognition systems can or should utilize acoustic information, which is normally filtered out in our auditory system, in order to obtain good results (see, e.g., *Hermansky*, 1997 for comments about auditory modeling in ASR). The choice of which parameters to use also depends substantially on the application. For example, the effects of noise caused by the transfer medium differ notably on the parameters used, and therefore many special methods for speech parametrization in noisy conditions have been developed (e.g., RASTA-PLP, *Hermansky et al.*, 1991; see the sub-section regarding linear prediction).

In practice there are an enormous number of interesting and exceptional ways to describe the speech signal in terms of parameters. While they all have their strengths and weaknesses, most of them will be bypassed here, and we will concentrate only on a few of the most used methods. These methods for feature extraction are based on the MFCC spectrum and LP algorithms. Also, direct use of the FFT spectrum is possible in ASR and it deserves a brief review, not least because the segmentation method presented in chapter 3 builds on it. These methods are described here in a short introductory manner, and for a more detailed description the reader should refer to, e.g., *Rabiner & Schafer* (1978) or *Motlíček* (2002). However, before going into parametrization methods, some fundamental properties of features will be discussed.

2.3.1 Philosophy of features

Classification is a process of naming a category for each entity to be classified. While we often refer to a description of *events* in the speech signal within the field of speech processing, a more general term called *item* will be used here to describe the phenomenon that we are trying to classify.

In order to categorize an item, a comparison or a type of distance evaluation has to be performed with respect to the other items that are to be categorized, or, to possible pre-defined categories available, in order to find the best categorical match for the item. Comparison and distance evaluation requires an estimate of the true value to compare and evaluate to, and the term *feature* is used to describe a piece of information that can be used for such a purpose. By comparing features we can measure the similarity (or difference) of two items, but as a very important distinction, only in terms of the categories that we are utilizing.

There is also an important link between features and patterns. While features are something that can be detached from the larger entity for more particular examination, a sufficient set of features can be considered as something that defines the entity. In cognitive and memory research this sufficient set is sometimes referred to as *defining*

features (*Neath & Surprenant*, 2003). In addition, there are *characteristic features* that do not directly define the concept, but are typical to that particular subject. One should immediately recognize that this division to two different types of features is not a strict property of the concepts, and depends on the context and the recognition process that is taking place. What is important is that structured (non-random) combinations, or sets, of features can be thought of as patterns by mapping them into the time domain. Especially in the case of human perception, the temporal dimension becomes an important mediator between features and recognition. Every activation process occurring in our neural system is a consequence of past activations and the effects of the environment. Without time there are no thoughts or imagery, no perception nor sensations. Every recognition process is a time sequence of consecutive neural messages originating from the stimuli, a pattern.

But what are good features for describing speech signals? Unfortunately, there is no universal theory or even unanimous rules of thumb for feature extraction in speech processing. While there have been some attempts to select a single best group of features for speech recognition (see, e.g., *Lee & Hwang*, 1996), none have proven to be feasible in all situations. The most central issues are with the nature of the items to be classified and the goal of classification that needs to be accomplished. It is also an important philosophical question whether to search for features that can be used to produce a categorization that fulfills the target expectations (based on prior knowledge), or whether to find features that categorize data in possibly unexpected but still systematic and beneficial ways. For example, in speech processing what most often is done is to make a separate two items that cannot be interchanged without affecting (the description of) the content of the signal, and at the same time, using features that are sufficient to describe the signal in terms of the ultimate goals. Often it is possible to exploit prior knowledge regarding linguistics, phonetics and audio signal processing in order to find satisfactory classification results that match the current conception of reality. For simplicity, this epistemological approach of exploiting pre-existing knowledge will be used for further classification discussion in this paper.

So as for classification, the first premise for a good feature is its ability to differentiate items from each other that are considered to belong into different categories in the best possible way. For example, in order to define whether an animal is a cat or a dog, we may not want to use the color of its fur as the most salient classification criteria, neither do we want to classify the gender of a person by considering only height. These features may be statistically inclined to point to a specific category, but as such they are not sufficient for making classification decisions. Preferable are those features that are in terms of order very distant from each other between the categories, and close to each other inside the category. In other words, good features should be *distinctive* and *contrasting* between the categories.

Usually in complex phenomena, one class of features is not sufficient for a desired classification, and therefore several different features are required. When several features are utilized, it is important to select the ones that contain complementary information concerning the classification problem. Also, when comparing features in some space with

specified distance metrics, the use of the mathematical term *orthogonality* to describe how much common information these features carry is preferred. Good features are as orthogonal as possible, meaning that they bring more information to the classification problem with minimum increase of complexity. By presenting more features with poor resolution capability to the classifier increases the complexity of the decision, and may hinder the overall decision quality.

The human perspective of feature detection is very closely linked to feature extraction in computational solutions. In everyday human perception, the most salient features are used as cues to help in restricting the possible categories searched for recognition, but they rarely lead to exact recognition. In order to obtain an accurate classification of an item or event, smaller and smaller supporting features must be exploited in addition to the most obvious ones to finally come up with only one most probable explanation (or categorical match). Therefore, recognition is also often considered as a hierarchical process. Several theories of hierarchical categorical memory have been established, and the interested reader is suggested to see *Collins* and *Quillian's* hierarchical model (*Collins & Quillian*, 1969), the feature overlap model (*Smith et al.*, 1974) and *Collins* and *Loftus's* spreading activation model (*Collins & Loftus*, 1975).

2.3.2 Linear prediction

Linear prediction (LP), sometimes referred to as *autoregressive analysis*, plays a central role in many speech coding, synthesis, and recognition applications. Its idea is, simply put, estimating filter coefficients for a filter that retains the waveform of the original or intended signal. LP is a model based on human speech production. It utilizes a conventional source-filter model, in which the glottal, vocal tract, and lip radiation transfer functions are integrated into one all-pole filter (see eq. 2.1) that the simulates acoustics of the vocal tract.

$$H(z) = \frac{G}{1 - \sum_{i=1}^{p} \hat{a}(i)^{-i}} \qquad (2.1)$$

In the above equation G is the gain of the system and $\hat{a}(i)$ are coefficients of a p:th order polynomial. The filter is driven by an excitation sequence $e(n)$ which is often referred to as the *residual* (*Deller et al.,* 2000).

Using an all-pole transfer function retains the magnitude properties of the signal but does not preserve phase information. While some components of speech can be better described with pole-zero models, motivation for using only poles comes from the formant structure of speech: especially in the case of vowels, the vocal tract can be considered as a long and relatively thin acoustic tube whose transfer function can be described sufficiently with an all-pole system. Resonance frequencies of the tube, referred to as *formants* in the case of speech, correspond to the LP-filter pole-pair locations in the frequency domain, allowing the filter to model the envelope of the frequency structure.

Also, the possibility to solve for the coefficients *â(i)* using relatively simple linear algebra is an advantage in all-pole systems. While the phase information of the signal is lost with this model, it is not usually significant since human speech perception is considered to be very insensitive to phase information in practice (*Moore*, 1995; *Karjalainen*, 1999).

The aim of LP-analysis is to find these so called LP coefficients *â(i)*. The core of the analysis can be thought of as in terms of minimizing the average mean squared error (MSE) $e(n)^2 = [s(n) - s'(n)]^2$ in the time-domain, meaning that the difference between the original waveform *s(n)* and the waveform produced by the source-filter system *s'(n)* is minimized for each windowed frame (eq. 2.2).

$$e(n) = s(n) - s'(n) = s(n) + \sum_{i=1}^{p} a(i)s(n) - i \qquad (2.2)$$

This problem setting can be inverted so that source-filter-output system becomes the output-filter-source system. Now the all-pole filter in eq. 2.1 becomes a finite filter (FIR) of length *p*+1 (eq. 2.3 and fig. 2.1).

$$\hat{A}(z) = 1 - \sum_{i=1}^{p} \hat{a}(i)z^{-i} \qquad (2.3)$$

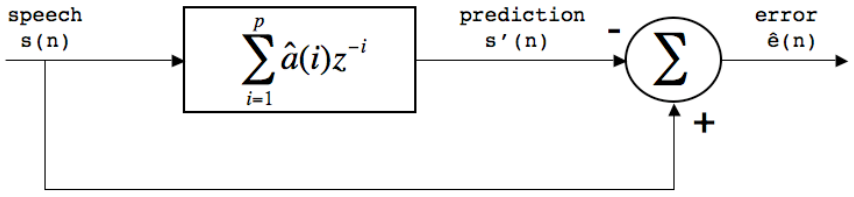

Figure 2.1: Inverse filter and prediction error (adapted from *Deller et al.*, 2000)

Attempts to minimize the residual *ê(n)* leads to *p* linear equations that can be solved in matrix form. After coefficients *â(i)* are determined, the gain *G* can be also calculated easily from the coefficients (see *Deller et al.*, 2000, for a more broad description of the mathematical derivations).

When linear predictive coding (LPC) is used for compression and transfer of speech signals, the residual, the gain and a few other possible parameters are packed with the LP-coefficients to enable reproduction of the original signal at the receiver. Signal reproduction (or LP-synthesis) takes place by using the above described source-filter system with *ê(n)* as the excitation signal and the *â(i)* as the coefficients specifying the filter transfer function *H(z)* (eq. 2.1).

Since linear prediction has been found to be a generally efficient way to parametrize speech, more refined versions of it have also been developed. One such version is

Hermansky et al.'s (1991) RASTA-PLP, abbreviation of *RelAtive SpecTrAl Perceptual Linear Prediction,* that has been specially designed to address the problems associated with noisy conditions. In addition to the usual auditory modeling contained in PLP (e.g., the use of a logarithmic critical-band spectrum and equal loudness conversion; *Hermansky*, 1990), each frequency channel is band-pass filtered by a filter with a sharp spectral zero at zero frequency. This suppresses constants and slowly varying spectral components from each critical-band, decreasing the effects of linear distortions caused by noisy communication channels. As a result, word error rates are significantly reduced in difficult conditions as compared to PLP.

2.3.3 MFCC

The use of Mel frequency cepstral coefficients can be considered as another standard method for feature extraction (*Motlíček*, 2002). This method reduces the frequency information of the speech signal into a small number of coefficients that emulate the separate critical bands in the basilar membrane of the ear, i.e., it tries to code the information in a similar way as the human cochlea does. Additionaly, the logarithmic operation attempts to model loudness perception in the human auditory system. MFCC is a very simplified model of auditory processing, but it is easy and relatively fast to compute.

Calculating MFCC coefficients consists of the following steps:

1. The signal is windowed with a specific window function (often Hamming or Hanning) using a window length of approximately 10-20 ms and a step size of 5-10 ms.
2. The spectrum is calculated for each window using the FFT.
3. The spectrum is then filtered with a special Mel-scaled filter bank to obtain corresponding Mel-coefficients. Single bands in the bank are usually triangular in shape, and overlapping each other.
4. The logarithm of Mel-coefficients is then computed.
5. The discrete cosine transform is used to transform into the cepstrum-space.
6. Non-necessary (high-frequency) MFCC-coefficients are discarded if desired.

The use of about 20 MFCC coefficients is common in ASR, although 10-12 coefficients are often considered to be sufficient for coding speech (see, e.g., *Hagen at al.*, 2003). The most notable downside of using MFCC is its sensitivity to noise due to its dependence on the spectral form, which has kept researchers searching for more robust methods to describe the speech signal. Methods that utilize information in the periodicity of speech signals could be used to overcome this problem, although speech contains also aperiodic content (*Ishizuka & Nakatani*, 2006).

2.3.4 Pure FFT

Despite the popularity of MFCCs and LPC, direct use of vectors containing coefficients of FFT power-spectrum are also possible for feature extraction. As compared to methods exploiting knowledge about the human auditory system, the pure FFT spectrum carries comparatively more information about the speech signal. However, much of the extra information is located at the relatively higher frequency bands when using high sampling rates (e.g., 44.1 kHz etc.), which are not usually considered to be salient in speech recognition. The logarithm of the FFT spectrum is also often used to model loudness perception.

The benefit of using a purely FFT-based approach is its linearity in the frequency domain and its computational speed. While it does not discard or distort information in any anticipatory manner, the representation of the signal remains easily perceivable for further analysis and post-processing. The effects of noise in the FFT spectrum can also be easily comprehended.

2.4 Data classification, a bridge to a top-down pathway

Data classification, often referred to as clustering, is a conventional follow-up process to feature extraction in many fields dealing with pattern discovery (*Jain et al.*, 1999). Clustering is often considered as a data classification problem, which can be used to impart understanding to complex phenomena that cannot be comprehended directly. The general aim of the analysis is to come up with a limited number of descriptions for the inspected data, which leads self-evidently to information compression and conversion.

In speech recognition tasks the usual aim is to group features, each group representing well-defined sections of the original speech signals, into semantically (words, sentences) or phonologically (phones, syllables) coherent groups. Recurring events, or patterns, in the speech stream can then be detected by statistically tracing the connection between the data and the clustering process (see, e.g., *Park & Glass*, 2006).

2.4.1 From clusters to memory

Clustering can also be paralleled to biological information processing. In humans, experience and past learning plays a large role in classifying incoming stimuli into specific (semantic) categories. The so called bottom-up and top-down interaction, partially driven by attention, tries to organize sensory input with internal representations in a way that the effectiveness of processing between environmental requirements and internal behavior becomes optimized. While it is too early to make any far reaching specifications about mechanisms behind this phenomenon, the statistical re-organization

of neural networks can be said to work as an infrastructure that provides researchers with a biological version of computation needed in information processing and learning (see *Buonomano & Merzenich*, 1998 for a description of cortical plasticity). Memory structures, as complex as they might be in reality, can be considered as mediators for matching operations between new entries of information and the existing knowledge base.

On a very abstract and conceptual level, human data processing is similar to speech data clustering in speech recognition systems, where finding the best or several good clusters for each vector is actually the process of matching new input with something already known. Similarly, expanding the cluster space by creating new clusters can be compared to learning, a process of building entries of new knowledge if they do not fit into any existing structure. Naturally, the clustering process may take place at several different levels, using different level data to build different levels of abstraction. However, in most of the current clustering methods used, the process is a deterministic purely bottom-up data analysis procedure, where all external knowledge has already been made implicit in the algorithm's instructions. The clustered data remains relatively static after the method has achieved the desired convergence, and the classification does not interact with, e.g., contextual information from the subsequent data frames, or, with external information that might or might not correlate with the data that is being clustered, unless all of the information is packed into the data itself. Of course, this is not the case with human pattern discovery where there are several sources of information available which may not occur exactly at the same time, or, in which some of the "data" may be actually refined products of initial pattern discovery processes taking place in other functional domains.

From the engineering point of view modeling dynamic and temporally divergent multimodal clustering operations may not even be reasonable in many cases since the complexity of the system may overwhelm congruent and analyzable ensembles. As has been done so far, it might be most practical to separate the process into several sub-systems, where clustering is one sub-system responsible for the process of classifying data provided by bottom-up processing. This information can then be utilized by other methods for more exquisite pattern discovery. Combining several operations into one large dynamic classification framework should not be entirely discarded though, as the developing knowledge and tools in the field may make possible novel methods for data processing.

2.4.2 Common approaches to data clustering

At the most fundamental level, the principle for creating the cluster space can be divided into two different categories: hierarchical methods and partitional methods (*Jain et al.*, 1999). In hierarchical approaches, each piece of input data, which is called a *data frame* here, is classified to a specific sub-level at each node, resulting in a *dendrogram* (fig. 2.2). The classification nodes may be distributed to several different levels, and each node can use specific distance criteria to evaluate the data. It is possible to use a different sub-space for each branch, or, all clusters may reside in the same n-dimensional space while the grouping is not performed with a direct distance metric evaluation in that space.

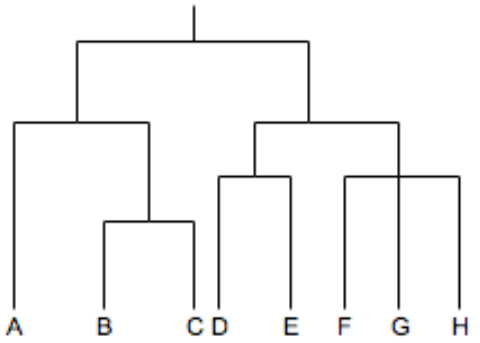

Figure 2.2: Dendrogram of hierarchical clustering.

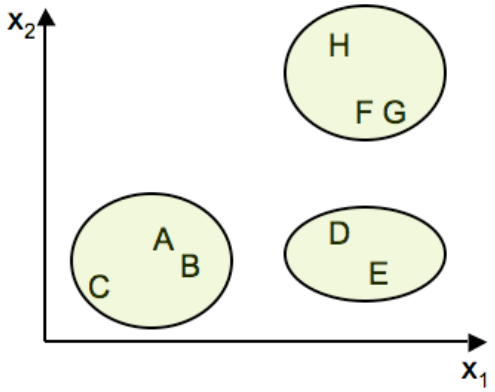

Figure 2.3: 2-dimensional mapping of partitional clustering.

In partitional clustering (fig. 2.3), the cluster space is usually created by dividing the data set into a number of clusters (spread around cluster *centroids*) by using a defined distance measure. The most commonly used partitional algorithm is *k*-means (McQueen, 1967), which starts with a random initial partition containing all of the data, and then reassigns the patterns to clusters until the convergence criterion is met. The name *k*-means comes from the squared error distance metrics, in which the final number of clusters is defined to be K. Formula 2.4 describes the squared error calculation for clustering ζ of pattern set $\aleph$.

$$e^2(\aleph,\zeta) = \sum_{j=1}^{K}\sum_{i=1}^{n_j}\left\|x_i^{(j)} - c_j\right\|^2 \qquad (2.4)$$

In the formula $x_i^{(j)}$ is the *i*:th pattern of the *j*:th cluster and c_j is the centroid of the cluster (*Jain et al.*, 1999).

Another common way to classify clustering methods is division to *agglomerative* and *divisive* methods, *hard* and *fuzzy* methods, and to *incremental* and *non-incremental* methods. In agglomerative methods, each data frame is placed in a distinct cluster and clusters are merged unless a termination criterion is met, while in divisive methods all data frames start in the same cluster that is then split up into several smaller clusters. Hard clustering methods assign each data frame into one best choice cluster, while in fuzzy clustering each frame has a degree of membership and may belong to several different clusters.

Especially in the case of ASR systems it is often important to differentiate between incremental and batch processing. Batch processing, in which large amounts of data are processed at a time, is not suitable for nearly real-time data processing as the feedback or results from clustering are needed before large amounts of data can be gathered. For bottom-up top-down interaction purposes, data classification must be performed for utterance size units at most, while the utterance should also be able to shape the cluster

space to facilitate learning. However, many of the clustering algorithms are designed to work with pre-existing sets of data (e.g., *k*-means), enabling the optimization of the entire cluster space when all possible information about the data is already available. This sets challenges for optimal classification in incremental situations where very little may be known about future input. However, in learning architectures like the one of which's foundations are being sketched in this thesis, batch processing can be used to simulate, e.g., rehearsal processes that take place during non-active periods of the system. To meet specific needs of incremental clustering a modified clustering algorithm will be presented in the next chapter.

2.5 Current approaches to top-down feedback in speech processing

The original idea of this section was to discuss the current status of unsupervised segmentation and speech recognition methods exploiting prior learned information. However, reality has indicated that the literature available in this area is nearly non-existent, as only very few ambitious attempts to model bottom-up top-down interaction through learning have been made so far. Several possible reasons for this may exist, but one central issue may be the sheer extent of the problem, spreading to numerous different fields of scientific research juxtaposed with the small quantity of real knowledge about effective memory and feedback processes. The first part of this section will be used for highlighting the importance of top-down processes. In the second part, two renowned computational memory models will be introduced, as they may help us to understand how a bridge between bottom-up processes and an intelligent learning agent could be built.

2.5.1 What is top-down feedback and how can it be exploited?

Motivation for the use of top-down information in segmentation is derived from human auditory processing. It has been shown that human speech perception combines bottom-up and top-down processing concurrently to make the best possible interpretation of incoming auditory signals (*Samuel*, 1981). As we already know, speech signals are often ambiguous and distorted in several ways in their acoustical form. Air as a medium can also carry only a limited amount of features encoded into a signal and therefore the recognition of intended messages is an extremely hard problem.

Top-down information can be thought to consist of two different types: the first type is the prior learned knowledge of the world, or in the case of speech perception, the grammar and lexicon of the language and the semantic framework that our working memory deals with. While we are listening to someone talk, our brain continuously analyzes the semantic content of the speech, compares the incoming signal to existing models of grammar, and performs word matching exploiting our lexical memory. In practice this appears as a good method that exhibits insensitivity to noise, to speaker dependent variability, and to distortions caused by the acoustic environment. It has been

shown that by masking entire phones using bursts of noise from otherwise semantically coherent sentences may not be perceived at all by test subjects (*Warren*, 1970). The defining aspects of this feedback are learning and memory, and their interaction with sensational processing. This is the type of processing that is usually meant when referring to top-down processing.

A second type of top-down information flow consists of lower level (sub-cortical) unconscious neural feedback systems. These systems can be considered as a type of "species built-in systems" that are the result of normal development of the individual, most of them being fully or partially operational in infancy. Many of these automatic processes are very similar in many mammalian species, especially for general (non-speech specific) auditory stimulae. The presence of one of these low-level feedback mechanisms in auditory processing can be demonstrated with otoacoustic emission (OAE) (*Kemp*, 1978). It is a phenomenon that can be observed when a short burst of sound is passed into the cochlea, and a faint echo of the stimulus can be heard a few milliseconds later. OAE is result of neural feedback and adaptation of the hair cells to the signal, and it is currently used to measure the health of the inner ear. Much of the structure of neural networks and their feedback systems in the central nervous system is still unknown, and there is not much knowledge about the role of sub-cortical processing in speech perception. However, the presence of tonotopic or even phonotopic maps in core areas of the auditory cortex express possibilities of some sort of discriminating speech processing at lower levels. These sort of low-level feedback mechanisms seem to be comparable to automatic control systems in engineering, where some features of the input signal define a group of parameters that are used to control the processing of the same signal (confer, e.g., *Ulfendahl & Flock,* 1998 for a description of effector hair cells that also play role in OAE).

To utilize such feedback mechanisms in artificial speech recognition systems, an understanding is required of what kind of information can be exploited directly from the speech signal (low-level feedback or adaptation), and what can be obtained by applying statistical methods to large amounts of spoken language (high level feedback, memory and learning). By directly inspecting the properties of an input signal, different built-in adaptation techniques may be used, e.g., adjusting the gain of the amplifier or coding of the signal (*Erdmann et al.,* 2000), which correspond to biological DNA-hard coded mechanisms aiding the perception of what the subject is provided with. For example, the stapedius reflex in the middle ear and other similar automated systems that generally facilitate processing in humans in many cases, but are not always appropriate or even useful in artificial systems. These sorts of mechanisms are the result of evolution and do not adapt to sudden changes in the environment. More importantly, this sort of simple input-process-adjust input -relation does not enable learning of patterns and structures that form language. It is merely manipulation of the form in which the data is presented.

As already pointed out, the general framework for human speech recognition relies heavily on learning. The better human's understand and speak different languages, the less problems they have in word recognition or comprehension even if the circumstances are not optimal (assuming the language is known). For foreign languages it is even

difficult to distinguish single words in utterances, but on the other hand even infants can rely on learned statistical cues to gain partial success in this task (*Saffran et al*., 1996). This supports the idea of hierarchical processing, where several different cues are combined to incrementally learn higher-level structures, in the end including grammar, lexicon and concepts associated with the language. As the language skills develop, automatic prediction and hypothesis testing become important aspects of speech perception. This is where bottom-up processing begins to provide cues for memory structures, but the information also starts to flow from top to down in order to test and refine these recognition hypotheses.

2.5.2 Computational models of speech perception

While the role of memory structures, spreading activation, and neural feedback in speech perception has not established a secure position in ASR, they have been of great interest in memory research. One well-known model attempting to model human speech recognition, TRACE (*McClelland & Elman*, 1986), blurs the line between perceptual processing and conceptual memory, integrating speech processing and primary memory into non-separable process. Despite its deficiencies, it manages to describe many aspects of the human speech recognition processes and effects that are central in memory research. It relies on small processing units extracted from the speech, which are interconnected with exhibitory and inhibitory connections. Units are organized into three separate levels, representing features, phonemes and words, respectively. Feature level units are produced from the speech with specific feature banks. Phonemes are then combined from a specific set of features, as the words become combined from phonemes. Horizontally connected units and vertically connected levels compete for best possible interpretation of the underlying utterance by using these principles, when each unit at each level represents a hypothesis of a corresponding unit in the speech. One major problem in TRACE is that it does not explain how sequences of phonemes can lead to the learning of new words. It relies on pre-defined structures to identify words, which in turn rely on hypotheses about phonemes formed by pre-defined features. Therefore, TRACE cannot be considered as a real high-level top-down feedback system, but as a slightly aided bottom-up model exploiting hierarchical spreading of activation. See *Neath & Surprenant* (2003) for a review of TRACE from the perspective of memory research.

From the perspective of bottom-up and top-down interaction, another interesting model is MINERVA2 (*Hintzman*, 1984; *Hintzman*, 1986). It is a simulation model that concentrates on the functionality of episodic memory, managing to represent concepts, or schema-abstractions, via activation of several entities during retrieval from the memory. A central concept of MINERVA2 is the memory trace: a list of features, where each element in the list represents one feature with integer values -1, 0 or +1. The memory model consists of primary memory (PM), which is responsible for sending probes (which are also traces) to the secondary memory (SM) to invoke memory retrieval, and for receiving trace echoes as a result of activation. As the probe is sent from PM to SM, SM peforms a direct similarity comparison between the probe and each trace stored in the memory, and results in the so called *activation* for each trace (which is the cube of the

similarity distance to provide some nonlinearity and to increase the SNR). By calculating the product of the activation level and feature values for each trace, and then summing the feature values of all activated traces, *echo content* is gained. Since some of the activated traces contain information that was not included in the probe, the *echo content* may differ from the original probe, which simulates associative recall. It is important to notice that the model relies on multiple-trace theories, i.e., each input repetition creates a new trace to the long-term memory instead of reinforcing the existing traces. Forgetting is simulated through a probability that some features of a new experience will be coded incorrectly, e.g., 0 may revert to 1.

Recently the modeling of episodic memory with the aid of MINERVA2 has been investigated in the area of speech recognition. Following the preliminary recognition work of *Wade et al*. (2002), *Maier & Moore* first used MINERVA2 for vowel recognition (*Maier & Moore*, 2005) and then for more extensive speech recognition (*Moore & Maier*, 2007). In vowel recognition the model achieved better results in comparison to state-of-the-art pattern classifiers, while the differences were not statistically significant. As for word recognition, the performance was found to be worse than methods using HMM. The authors suggests that this is due to the inability of the model to utilize temporal sequences, and also the effect of using MFCCs with the Euclidean distance might hamper the process. However, the advantage of MINERVA2 in comparison to HMMs is that it retains the fine phonetic details instead of averaging them out to represent single states (*Moore & Maier*, 2007).

Interestingly enough, most speech recognition attempts with computational models of memory are working directly with features extracted from the acoustic signal. In practice this means that there is no pre-classification for the input, nor is there a way to capture temporal aspects of the signal unless the model contains built-in properties for temporal activation (which is not very well represented in MINERVA2). The models are working directly on the coefficients taken from each time window, which denies the use of context to help in recognition. By adding an intermediate processing level to convert acoustic features into sequences of segment classifications, could help to capture the temporal aspects of the speech, as well as the acoustical information for each meaningful unit, in the end improving the performance of the memory and recognition functions. One way to achieve this would be to use segmentation and segmental clustering as a front-end processing method.

3 Constructing the bottom-up process

This thesis was written while developing a new architecture for speech recognition systems that tries to simulate the language acquisition processes of a child. Most of the work, methods, and results were conducted while working within an international research project called "Acquisition of Recognition and Communication Skills" (ACORNS[4]), funded by the European Commission. One of the main targets for the project is (at the time of writing) the development of methodology and tools for an interactive memory prediction framework where bottom-up processing is coupled with memory functions that can provide top-down feedback for speech recognition. Not only the recognition of speech as a whole, but also segmentation and classification processes themselves, have a large potential for improving their results when utilizing memory structures that may provide the algorithm with higher level information concerning the processed input. This information can be grammatical or lexical, it may provide clues extracted from the prosody of the sentence, or it may use multimodal information for associative pairing. In a finalized form, the system may even be able to make predictions regarding the incoming input.

An important point in the memory-prediction framework is that the system must learn something before it can be used for recognition[5]. In humans, without learning and daily life experiencing there is no knowledge, language, nor memories. There are only genetically hard-coded mechanisms for perception (e.g., ears), for processing the sensory input in our brains (the neural auditory system associated with all other functions in our brains), and a possibility to respond to our environment with motoric actions (including speech production). In a similar manner, the computational speech recognition system under development does not "know" anything beforehand. Therefore, it will need mechanisms for converting the signal inputs into representational forms that can be processed statistically to form "memories" and ultimately enable learning. In the rest of this thesis, potential mechanisms for providing such bottom-up information for higher levels are introduced, setting the foundation for future more extensive memory-prediction frameworks that are currently beyond the scope of this work and awaiting further research.

[4] http://www.acorns-project.org

[5] Recognition, re+cognition, refers to the notion that some familiar semantically comprehensible item recurs. In other words, the external world provides sensory input that has similar properties (patterns?) to something already known by the observer.

In this chapter a novel algorithm is introduced for blind speech segmentation that does not exploit any prior nor external knowledge of the speech signal. Segmentation will be coupled with a feature extraction method in order to feed phone-level data to higher processing levels. To build some effective connection between the bottom-up and top-down processes, different approaches to clustering of segmental data will be also reviewed. We will also shortly dwell on the evaluation methodology of these processes.

All practical implementations of the algorithms and the experiments reported in this work were carried out in the MATLAB-programming environment[6].

3.1 Bottom-up processing of speech signals

The front-end of bottom-up processing is split into two parallel processes: a blind segmentation of speech into phone-sized units, and a feature extraction (FE) process to provide descriptive data for each segment. Both methods are based on the Discrete Fourier transform (DFT) representations of the signal. The reason for this division lies with the slightly differing needs of these two processes: the segmentation algorithm needs to find robust and salient cues that help to estimate possible phone boundaries, while feature extraction has to build congruent and normalized parametric representations of the segments that describe each phone in some best possible manner. Making these two processes independent of each other also allows greater flexibility from the development perspective, and can be reasoned from the cognitive point of view as well, e.g., the spreading of stimulus based activation into several different neural networks for processing.

The classification process of the segmental data will also be considered through the use of data clustering. Clustering is an important step occurring between bottom-up processing of raw data and the statistics that are built upon it. Two different methods with several variations were tested for their functionality, while the most efficient manner to perform clustering still remains an open question. Furthermore, the statistical methodologies that will be built upon clustering will impose their own requirements to the clustering process. To facilitate further work with this classification and pattern discovery problem, two different approaches to segmental data clustering will be presented here as potential candidates.

3.1.1 Bottom-up algorithm for segmentation

The segmentation method described here is a purely bottom-up blind speech segmentation algorithm, which does not utilize any external information besides parameter estimation during its development. It produces a phone-level segmentation of the speech signal with a probability classification for each segment boundary. The

[6] MATLAB © The MathWorks, Inc., http://www.mathworks.com

general principle of the algorithm is to track the spectral changes in the signal by comparing the cross-correlation of the FFT coefficients as a function of time, and to place segment boundaries at the locations where spectral changes exceed a minimum threshold level.

Both the segmentation and feature extraction algorithms take one speech signal waveform at a time as an input. The signal is pre-emphasized with a typical 2nd order pre-emphasis FIR-filter of the form

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \qquad (3.1)$$

in order to emphasize formant frequencies compared to very low and very high frequency information. Values for constants b_0 = 0.3426, b_1 = 0.4945 and b_2 = -0.64 are used (see Appendix A for phase and frequency response). This was found to improve segmentation results slightly as compared to a regular 1st order FIR, which may be due to the fact that the 2nd order filter suppresses the effects of high-frequency spectral changes that may occur during, e.g., fricatives, and therefore cause insertions due to phone splitting.

After pre-emphasis, the absolute value DFT is calculated from the signal using a short 6 ms Hamming window with a 2 ms step size using the Fast Fourier transform (FFT). Motivation for the relatively short window stems from the properties of pitch-periods: a best possible spectral representation of the short-term signal is desired, so by selecting window sizes approximately one pitch-period in length and centering the energy of the glottal pulse in the central part of the window results in a good spectral contrast (coefficients have the largest possible deviation from zero). By using very short window steps (~2 ms), the window will often be in synchrony with each pitch period, and the complex process of tracking pitch periods to gain pitch-synchronous windowing is avoided (so called *pitch-synchronous analysis*).

The zero-crossing rate (ZCR) and a logarithm of the short-term energy (STE) of the signal are then calculated from each window and stored for later use. Each frame is further compressed in the frequency-domain with an asymmetric tanh[x]-mapping (see Appendix A) to emphasize the formant information in the spectrum. Also, the mean power of the each frame is subtracted. All spectral frames are stored into a matrix **M.** After the entire signal has been processed, a cross-correlation matrix **C** is calculated from the **M** (see fig. 3.1). The matrix **C** now represents the inner coherence (correlation) of the speech signal as a function of time. When visualized, bright areas represent parts of the speech that are highly coherent (high correlation with neighboring frames), while darker areas represent changes in the spectrum. The diagonal (seen as a thin white line in the figure) is linear with respect to time, running from the upper left corner (the beginning of the signal) to the bottom right corner (the end of the signal).

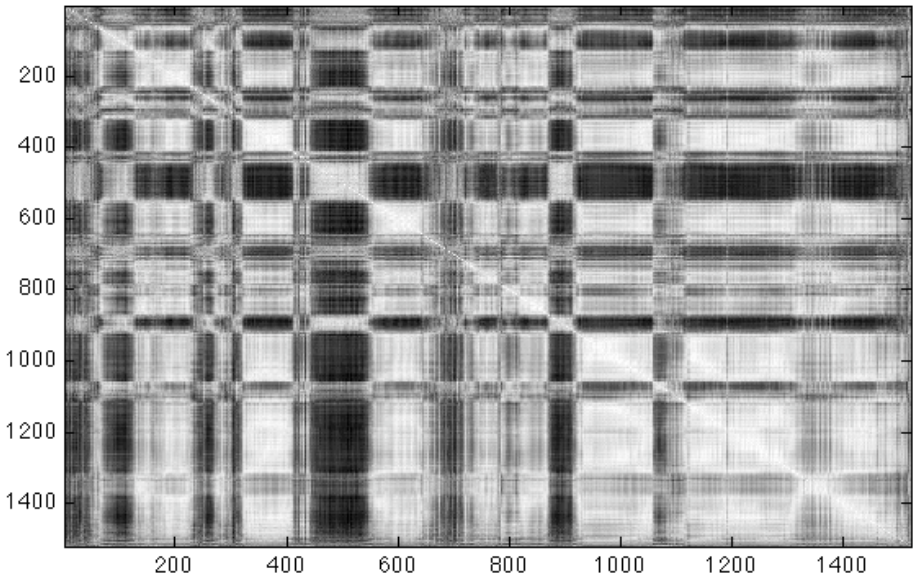

Figure 3.1: Cross-correlation matrix **C** calculated from spectral vectors.

Next, a special 2-dimensional filter is used to obtain information from matrix **C**: a square of size d_1 x d_1 and two triangles with sides of length d_2 are aligned next to the diagonal of the matrix (fig. 3.2).

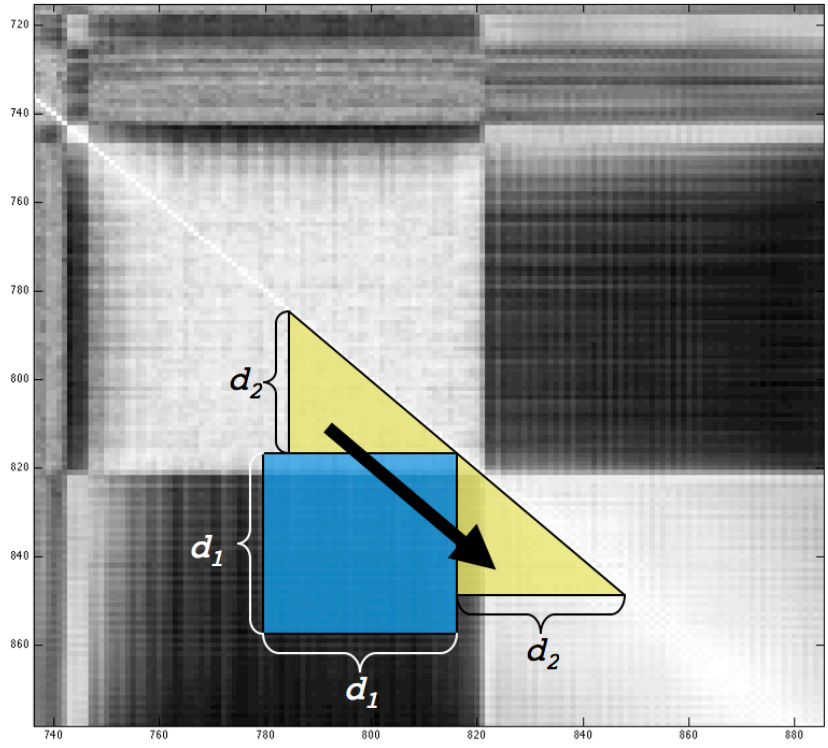

Figure 3.2: A zoomed-in visualization of part of a correlation matrix. The 2D-filter moves along the diagonal.

As this integrator slides along the diagonal, i.e., the time-axis, from corner to corner, the sum of the elements under the triangles t_s[m] is subtracted from the sum under square b_s[m] at each moment in time (eq. 3.2).

$$s[m] = t_s[m] - b_s[m] \tag{3.2}$$

This produces a representation s[m] of the speech signal where large negative peaks reflect large spectral changes and point to possible segment boundary locations (fig 3.3). The resolving capability of s[m] can be adjusted by changing the parameters d_1 and d_2, which is basically a trade-off between the temporal accuracy and boundary detection reliability.

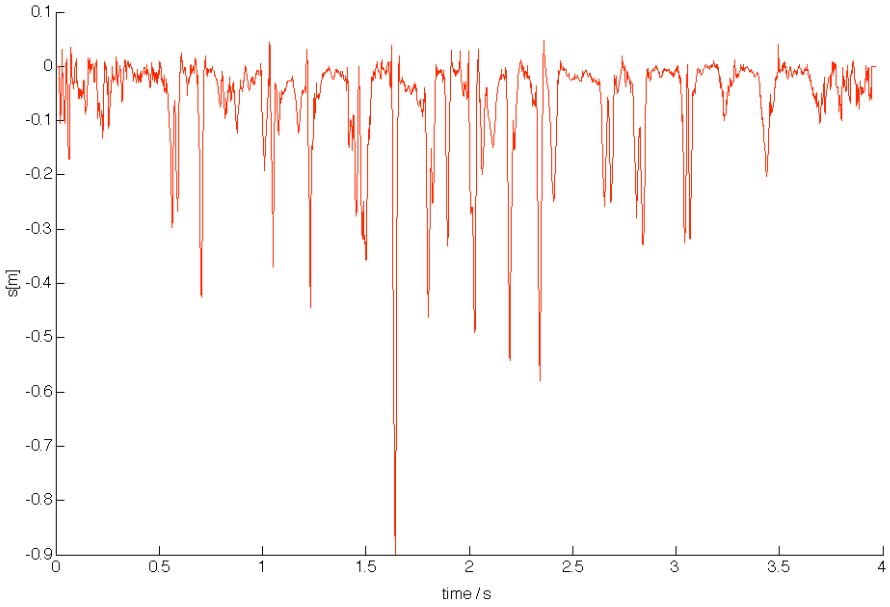

Figure 3.3: Signal s[m] produced by sliding integrators.

The signal s[m] itself is rather noisy, so another special filter, a so called *Minmax*-filter, is used to refine the representation. As the filter passes through the signal, at each point it takes n_{mm} subsequent samples from s[m] and determines the maximum v_{max} and minimum v_{min} values of this vector. The filter produces the difference d_{max}=v_{max}-v_{min} as an output to the point where the minimum value was located (note that deep valleys in s[m] were hypothesized segment boundaries). The following pseudo-code illustrates the functionality of the filter (see also Appendix A for a MATLAB realization of this filter):

$$\begin{aligned} d_{\max} &= \max(s[m:m+n_{mm}]) - \min(s[m:m+n_{mm}]) \\ I &= find(\min(s[m:m+n_{mm}])) \\ s'[m+I] &= d_{\max} \end{aligned} \tag{3.3}$$

As a result of filtering, signal s'[m] is obtained (fig 3.4), in which the estimated segment boundary locations are now represented as easily perceivable positive peaks. Peak heights

are normalized to a scalar value ranging from 0 to 1 to provide a probability classification for each boundary: the higher the peak, the larger the local change in the spectral properties and the more probable it is that a phone transition has occurred.

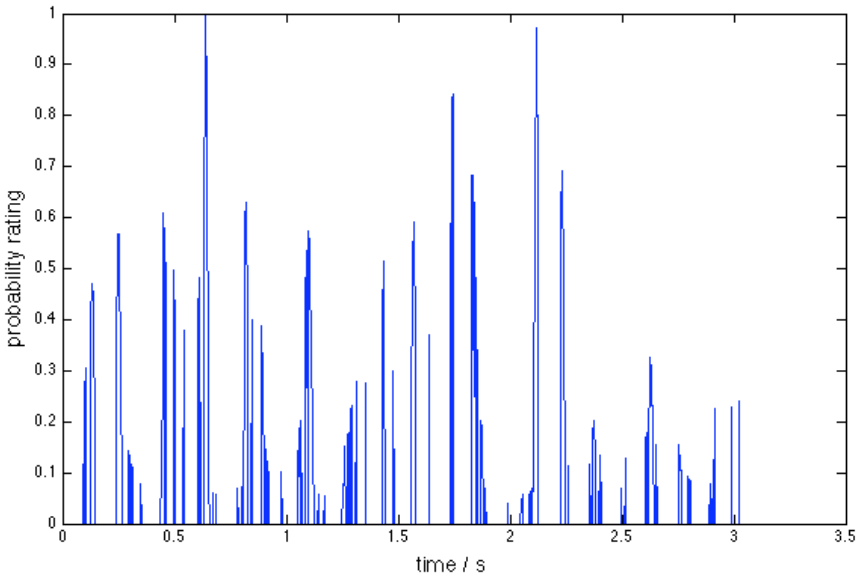

Figure 3.4: *s'*[m] created by Minmax-filtering from *s*[m].

Especially in the case of long spectral transitions between two separate phones, there may be several peaks very close to each other. To further refine the segmentation accuracy and to avoid over-segmentation, another special non-linear method is applied to boundary detection: in the time domain a distance between each peak crossing the threshold p_{min} is calculated. If two or more peaks are closer than t_d to each other, the probability ratings of the peaks are compared. Only the most probable (the highest) peak is retained, while its location is slightly adjusted towards the removed peak(s). The new location is between the old peaks and directly proportional to the ratio of probability ratings of these peaks in question. As a result, a refined s_r[m] is obtained.

In theory, a list of found segment boundaries $\boldsymbol{b_f}$ can then be created by choosing all of the peaks that cross the peak probability classification threshold p_{min}. In practice it is not usually necessary to segment the silent portions of the signal into several segments. This can be avoided by comparing the energy of the original signal during each peak to a minimum energy threshold e_{min}. Also, the ZCR can be used in addition to the energy for silence (and potentially fricative) detection. After all peaks passing the thresholds are chosen, locations of the segments in the list are then converted back to the time domain to gain boundary locations in seconds.

The structure of the entire segmentation process is depicted in fig 3.5 as a block diagram.

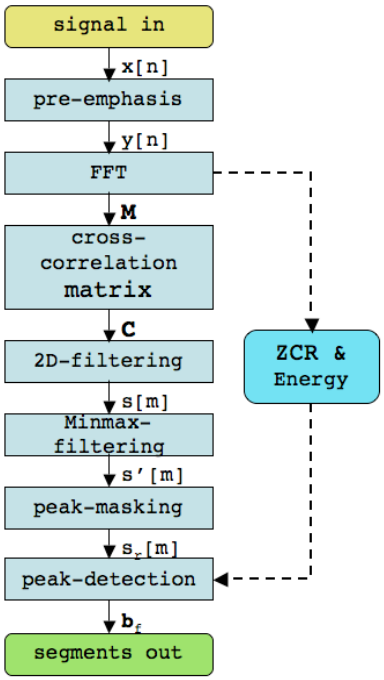

Figure 3.5: Structure of the segmentation algorithm.

3.1.2 Algorithm for feature extraction

The purpose of the feature extraction algorithm is to provide a coherent representation of the segments in a way which different phones have non-equivalent qualities and similar phones have equivalent qualities. In practice, there are numerous variations, or allophones, for every phoneme depending on the context and speaker, and therefore classification to strict phonemic categories can not be accomplished with this type of blind methodology. Consequently, the feature extraction algorithm creates representations of the segments with self-contained heuristics for spectral descriptions without any true knowledge of the underlying signal. Feature extraction and subsequent data classification will rely on the FFT representation of the segments to classify the speech data.

The feature extraction method uses the same 2nd order pre-emphasis and 6 ms Hamming windowed FFT as the segmentation algorithm to produce n spectral coefficients for frames each 2 ms in length. In addition to the short-term energy, the variance of the spectral coefficients is also stored for each frame. The tilt of the spectrum is removed along with the average power of the signal and both are stored for later use. Finally, each frame vector containing the spectral coefficients is normalized to a unit vector and stored into a matrix **F**.

Now a segment boundary listing $\boldsymbol{b_f}$ of length N_b, provided by the bottom-up segmentation algorithm introduced above, is taken as the input. Using the boundary location information, matrix **F** is then divided up into N_b-1 segments. Furthermore, each segment

is split into onset and offset parts, the former being the first d = 40 % of the segment, while latter covering the last 60 % of the duration. Small (default m = 10 %) margins m_1 and m_2 are assigned to the edges of the segment to attenuate the effect of small displacements in the segment boundaries and to avoid using very ambiguous data from the middle of phone transitions between two phones (fig. 3.6).

First 40 %		Last 60 %	
m_1	Onset	Offset	m_2
10 %	30 %	50 %	10 %

Fig. 3.6: Segment division into onset and offset parts with corresponding default values.

The 40/60-division is motivated by the observation that in most voiced phones articulation of the phone reaches its *locus*, or the maximum of articulatory "purity", during the first 40 % of its duration. For example, in vowels this means temporally relatively stable formants. After reaching the locus of the phone the articulators start to prepare for the following phone causing more context-dependent effects noticable in the spectrum, or, so called *co-articulation effects* (see fig. 3.7). In order to find normative but context-independent descriptions for segments, co-articulation effects should be avoided. However, for speech recognition purposes one might also require the knowledge of which context each phone appears in. For that purpose a spectral representation of the segment from the latter part is extracted. The effects of this preliminary estimate for segment division will be investigated in experiments described in the next chapter.

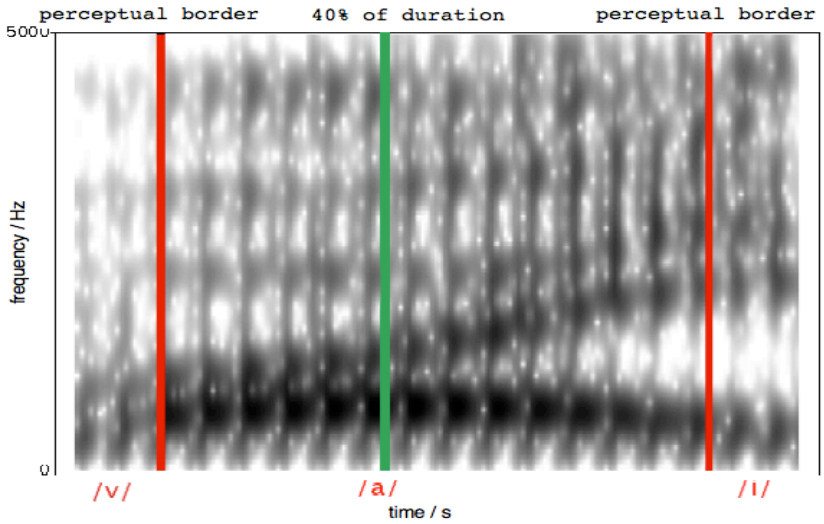

Figure 3.7: Adjacent phones /a/ and /i/ from a Finnish speech sample "*vaippa*". The green line indicates the location of 40 % duration of /a/, while the red borders indicate perceptual transitions from /v/ to /a/ and from /a/ to /i/.

The next step is to create a spectral vector that represents the spectral structure of a segment. To gain a good representation of both parts of the segment, only spectral vectors containing a good spectral contrast are chosen from **F**. In principle, these good contrasting vectors are a result of a pitch-synchronous windowing. The algorithm finds these vectors by inspecting the variance of the spectral coefficients during each frame

(see fig 3.8). A maximum contrasting frame is indicated by a maximal deviation of the spectral coefficients from the zero level, which in this case is the frame with the highest local variance. Again in practice, it is possible to estimate the variance from either the transformed spectral coefficients or from the windowed signal before the FFT. The maximums are located at identical locations in both cases.

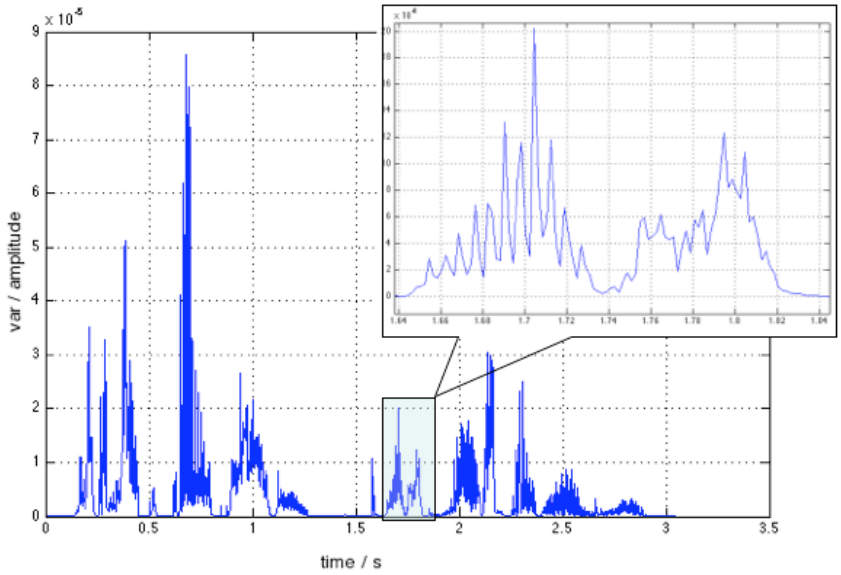

Figure 3.8: Variance of the spectral coefficients calculated for each 6 ms window with 2 ms steps. Pitch-synchronous frames can be seen as peaks that are spaced evenly approximately 8 ms away from each other.

The algorithm selects n_s (default n_s = 5) best contrasting spectral vectors for both the onset and offset parts of the segment, and averages them into two corresponding spectral vectors $\boldsymbol{sv}_{on,i}$ and $\boldsymbol{sv}_{off,i}$. Signal energies related to these vectors are averaged into $e_{on,i}$ and $e_{off,i}$, average variance of the spectrum over the whole segment is averaged into var_i. In addition, voicing (or lack of it) of the segment is estimated by a standard cepstral analysis and presented as a binary v_i = 0 / 1 using a pre-defined threshold (according to *Ladefoged*, 1982). Finally, the duration of the segment is included. Together they form an entity, which we shall call a *feature vector* $\boldsymbol{fv}_i$ (fig. 3.9) that contains all the necessary[7] information about the segment.

Figure 3.9: Feature vector $\boldsymbol{fv}_i$ created from the i:th segment of the input signal.

Finally, each feature vector created from the speech signal is stored into a matrix **S** for later use. Figure 3.10 shows the overall structure of the feature extraction algorithm.

[7] Necessary refers here to those features that can or will be utilized in the data classification problem in the clustering algorithm or in the analysis of the functionality of the algorithm. It does not include the assumption that these are the right and only features needed to provide the best possible segmental description for classification.

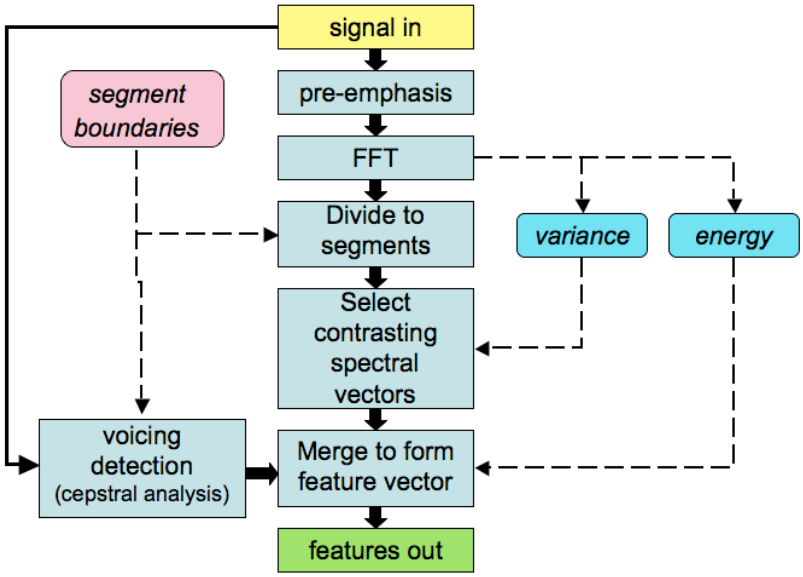

Figure 3.10: Structure of the feature extraction algorithm.

3.1.3 Methodological approaches to clustering

An important aspect in the clustering of segmental data in the framework of learning and interacting systems is the need for incremental processing. This sets constraints on the possible choices for a clustering algorithm. Several varying approaches were tested in order to gain a better understanding of the segmental data and the behavior and effect of the features used. Two relatively simple approaches were incorporated to the system represented in this study, and they will be introduced in this section. Their performance will be analyzed and discussed in the next chapter. The clustering algorithms represented here are considered as speaker independent without any speaker normalization.

The general idea in both methods relies on the incremental construction of the cluster space: the cluster space starts out as empty. Once the first utterance is segmented and features extracted, the feature vector $\boldsymbol{fv_1}$ (see the last section for a description) is used to form a first cluster χ_1. Then feature vector $\boldsymbol{fv_2}$, the description of the second segment in the utterance, is selected and the distance $c_{d2,1}$ between the spectral coefficients of the vector and the cluster χ_1 is calculated with a selected distance metric. If the distance is smaller than a merging threshold (in the case of this study, cross-correlation is larger than the threshold t_m), and all additional criteria required for the similarity are met, then $\boldsymbol{fv_2}$ is merged into χ_1. In the merging process a new centroid will be calculated for the cluster by averaging the new feature vector into the existing cluster centroid with a weight $1/N$, where N is the total number of vectors merged into that cluster including the currently incoming one. If the distance c_d is too large, a new cluster centroid χ_2 will be created. This process will be repeated for each created segment on a one by one basis, and utterance by utterance as well. Appendix A contains a pseudo-code description for the algorithm.

It is important to note that once the feature vectors are merged into clusters, their detailed information is discarded, i.e., there are no "clusters" for several single data frames in a literal sense, but only the centroids of the clusters are stored. This is a practical solution that addresses computational limitations since the amount of data would otherwise grow linearly as the system processed speech; practical systems should theoretically be able to classify unlimited amounts of speech without running out of memory. See appendix A for an enlightening demonstrative calculation.

The simplest principle for separating different phones from each other would be to compare the structure of their spectrum. Therefore, the basic distance metric used in clustering in this study is the cross-correlation between spectral coefficients, which can be easily obtained for the normalized spectral vectors by using a scalar product

$$c_d = \mathrm{A} \cdot \mathrm{B} = \sum_{i=1}^{L/2} a_i b_i \qquad (3.4)$$

where L is the FFT window length, thereby $L/2$ being the number of spectral coefficients (recall the symmetrical properties of the spectrum). The main difference between the two clustering algorithms used is the nature of the cluster space: in the first method, which shall be called a *Single-space Method*, contains only one large cluster space where each cluster centroid is located. All feature data will be incrementally added to the existing clusters or used to form new ones if the merging criteria are not met. Despite the name of the algorithm, this single space can also be divided up into several sub-spaces by adding additional merging criteria. By using, e.g., a binary decision for segment voicing, it is possible to isolate voiced and unvoiced segments from each other. The distance from the vectors to the clusters is calculated by using all of the $n = L/2$ spectral coefficients of the segment.

The second method, which shall be called a *Multi-level Method*, differs from the former in that it contains several hierarchical levels (fig 3.11). Each level uses a different portion of the spectral data as a merging criterion: one level compares the correlation of the spectrum in the 0-1000 Hz band, indicating mainly a location for the first formant in voiced segments. Another level uses the 1000-2000 Hz band as a comparison criterion, and the last one compares the 2000-3000 Hz band. The hierarchical order of these three bands is not fixed, and the performance of each combination will be evaluated in the experiments. The motivation behind band separation is the well known observation that the first two formants are usually sufficient to distinguish different vowels from each other (*Peterson & Barney*, 1952). The first formant usually lies in the frequencies below 1000 Hz, while the second formant moves mainly between 1-2 kHz[8]. Therefore, classifying the segments by the positions of these formants can be considered reasonable.

[8] *Parikh & Loizou* (2005) used a similar division to low- and middle-frequencies in a noise analysis of formant recognition, differing in that they used 1-2.7 kHz for the second band.

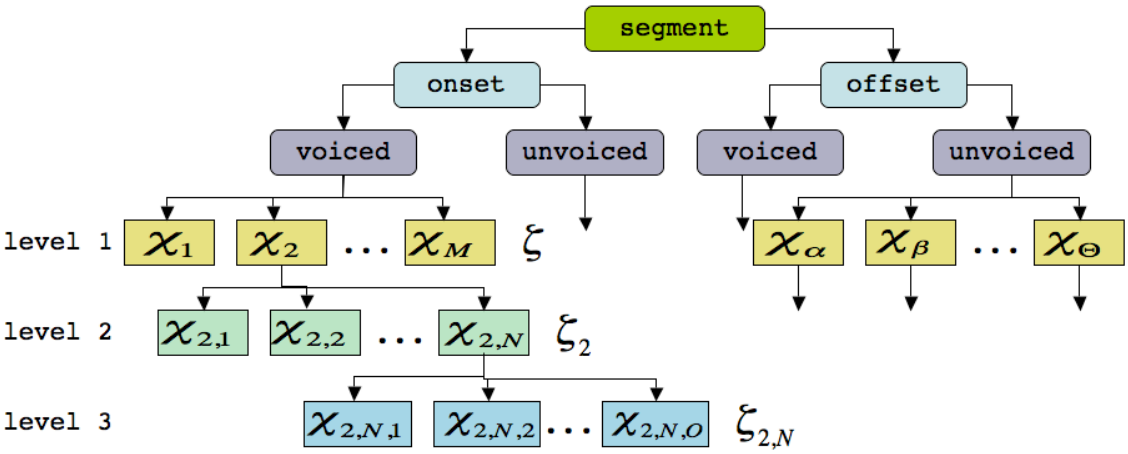

Figure 3.11: Block-diagram for the *Multi-level Method*.

In the hierarchy, there is yet again a possibility for making a division between voiced and unvoiced segment-spaces. Also, the onset part vectors and the offset part vectors of the segment can be segmented to different spaces, the first one containing statistics about the "stable" nature of the segments, and the latter having information concerning the context due to the co-articulation effects. The advantage of the hierarchical structure can also be seen by the flexibility of classification decisions that can be made at different levels, enabling easy experimentation with different decision criteria. It is also possible to explore the clustering results at different levels, and also to have several segmental "labels" for the same data at the same time with different classification precisions. The hierarchical system also reduces problems with computational complexity as the classification process can be partitioned into smaller sub-problems, reducing total memory and CPU-time requirements. Evident disadvantages of the hierarchy are its susceptibility to gross classification errors (e.g., a wrong branch selection at the first level) and the difficulty of uniform management of the data across the several subspaces. The fundamental clustering process of both algorithms is shown below:

Single-space Method

1. Take the spectral coefficients $\boldsymbol{sv}_{on,i}$ for a segment onset OR $\boldsymbol{sv}_{off,i}$ for a segment offset from the corresponding feature vector $\boldsymbol{fv}_i$.
2. Calculate a distance $c_{di,j}$ to all clusters χ_j, $j = [0,\ldots,N_j]$ in the same space ζ.
3. If the smallest distance $c_{di,m} \geq t_m$, merge the vector to the cluster χ_m (note that larger correlation means more similarity). If $c_{di,m} < t_m$, create a new cluster centroid with the properties of $\boldsymbol{fv}_i$.
4. Repeat for all N_i segments in the utterance

Multi-level Method

1. Take the spectral coefficients $\boldsymbol{sv}_{on,i}$ for a segment onset OR $\boldsymbol{sv}_{off,i}$ for a segment offset from the corresponding feature vector $\boldsymbol{fv}_i$.
2. Calculate a distance $c_{di,j}$ to all clusters χ_j, j = [0,...,M] in the same space ζ by using a frequency range B_1 = [f_{11} f_{12}] of the coefficients.
3. If the smallest distance $c_{di,m} \geq t_m$, merge the vector to the cluster χ_m. If $c_{di,m} < t_m$, create a new cluster centroid χ_{M+1} with the properties of the $\boldsymbol{fv}_i$ to this space ζ. Also, create a new subspace ζ_{M+1} for χ_{M+1} (level 2) and also create a copy of cluster $\chi_{M+1,1}$ there. Repeat this for level 3 to create $\chi_{M+1,1,1}$ in $\zeta_{M+1,1}$.
4. If the vector was merged at level 1 to the cluster χ_m, replicate the vector to the subspace ζ_m at level 2 and calculate a distance $c_{di,n}$ to all clusters $\chi_{m,k}$, k = [0,…,N] in the same space by using a frequency range B_2 = [f_{21} f_{22}] of the coefficients.
5. If the smallest distance $c_{di,n} \geq t_n$, merge the vector to the cluster $\chi_{m,n}$. If $c_{di,n} < t_n$, create a new cluster centroid $\chi_{m,N+1}$ with the properties of $\boldsymbol{fv}_i$ to this level and create $\chi_{m,N+1,1}$ to the third level sub-space $\zeta_{m,N+1}$.
6. If the vector was merged at level 2 to cluster $\chi_{m,n}$ go to the subspace $\zeta_{m,n}$ at level 3 and calculate a distance $c_{di,o}$ to all clusters $\chi_{m,n,l}$, l = [0,…,O] in the same space by using a frequency range B_3 = [f_{31} f_{32}].
7. If the smallest distance $c_{di,o} \geq t_o$, merge the vector the cluster $\chi_{m,n,o}$. If $c_{di,o} < t_o$, create a new cluster centroid $\chi_{m,n,O+1}$ with the properties of the $\boldsymbol{fv}_i$ to this level.
8. Repeat for all N_i segments in the utterance.

Especially in the *Single-space Method* there is a necessity for auxiliary functions that are used to "overhaul" the cluster space. The incrementality and non-predefined number of clusters leads to a severe expansion in their number with higher correlation radius thresholds. New data and noise always produce numerous new clusters consisting of only one or two segments. Also, the cluster centroids move in the space due to the averaging process with a speed proportional to the correlation radius of the cluster and to 1/N, where N is the number of segments in the cluster. When the cluster grows larger, its location starts to converge. However, movement during the first few segments may cause it to move inside the radius of another cluster. Therefore, so-called space *cleaning* operations are needed: clusters with size less than n_{merg} are merged to the closest cluster. Also, larger clusters that are too close to each other ($c_{ij} > t_{merg}$) will be merged together. The cleaning operation is repeated within specified intervals.

Since the movement of the cluster centroids and the merging and cleaning operations are not self-explanatory processes from the perspective of the classification problem, their effects are explored in experiments. Also, so-called integrative merging will be tested. In integrative merging, the first N_{max} segments arriving to the cluster are averaged normally with a 1/N weight. After the limit N_{max} is reached, new segments will be averaged into the cluster with a constant weight instead of *1/N*. This allows small but continuous centroid

movement over indefinite amounts of speech material, which may have effects on how the cluster spaces become populated. It is also a common technique found in the literature to adaptively adjust the radius of a cluster as the number of segments in the cluster increases (e.g., *Kim et al.*, 2005). The effects of increasing the selectivity as the cluster becomes larger were also briefly experimented with.

3.2 Evaluation methods

For development and testing a speech processing system, reasonable and descriptive metrics are needed in order to evaluate the quality of the work. While there may be one or more final goals that the system should achieve in a best possible manner, intermediate evaluation is also a necessary process in order to compare different approaches and to justify selected methods and parameters.

3.2.1 Evaluation of segmentation quality

In order to evaluate the quality of segmentation, it is necessary to have a reliable reference for phonetic segment boundaries. In these experiments, one method for the evaluation of the segmentation follows a literature convention of manual annotation comparison, in which segmentation boundary locations are compared against the manual annotated boundaries. While manual segmentation is prone to variability of individual judgments, it can be thought of as a reliable baseline for quality if it is sophisticatedly produced (*Wesenick & Knipp*, 1996).

A basic evaluation method promoted in the literature is a simple boundary distance comparison. Boundaries from the annotation are compared to the boundaries produced by the algorithm. *Insertions* are produced when there are boundaries created by the algorithm that do not match any annotated boundary, or if there are several boundaries produced in the vicinity of only one reference boundary. *Deletions* are produced when there is a boundary marked in the reference, but the algorithm produces no corresponding boundary. Also, found boundaries are considered as *hits,* and sometimes deletions are referred to as *misses*.

As simple as it may first seem, problems emerge when it is considered how actual hits are counted. Some papers (e.g., *Aversano et al.*, 2001; E*stevan et al.*, 2007) use methods defined in *Petek et al.* (1996, see below), while some others (*Sarkar & Sreenivas*, 2005) use basically the same principles without referring to any existing methodology in literature or without describing their methods of evaluation in detail. The definition of the method in the *Petek et al.*'s paper is the following: a search region $\mathbf{r_1}$ of size 40 ms is taken and placed symmetrically around a reference boundary (see fig. 3.11). Then it is checked if there are any boundaries inside that time window that are produced by the segmentation algorithm. *"If no peaks [of the algorithm] can be found in a search region,*

a deletion is detected", (*Petek et al.,* 1996). If there is more than one peak in the search region, these additional peaks are counted as insertions.

However, the definition above or in later papers adapting this methodology, do not describe what happens when the next reference boundary is examined and its search region $\mathbf{r_2}$ overlaps with the previous $\mathbf{r_1}$: are boundaries being searched again from the entire region $\mathbf{r_2}$, without concern for which peaks were defined as insertions in the last region? If it is possible to re-use algorithm boundaries from the previous search region, less deletions are obtained as many of the (reference) boundaries are located closer than 40 ms apart due to the nature of speech. Also, when a boundary in the last region $\mathbf{r_1}$ is defined as an insertion, it may actually offer quite a good match for the next reference boundary in the middle of $\mathbf{r_2}$. If it is permitted to use a little bit of common sense, it may concluded that the boundaries considered as insertions in the last region can still be used to create a hit in the subsequent frame.

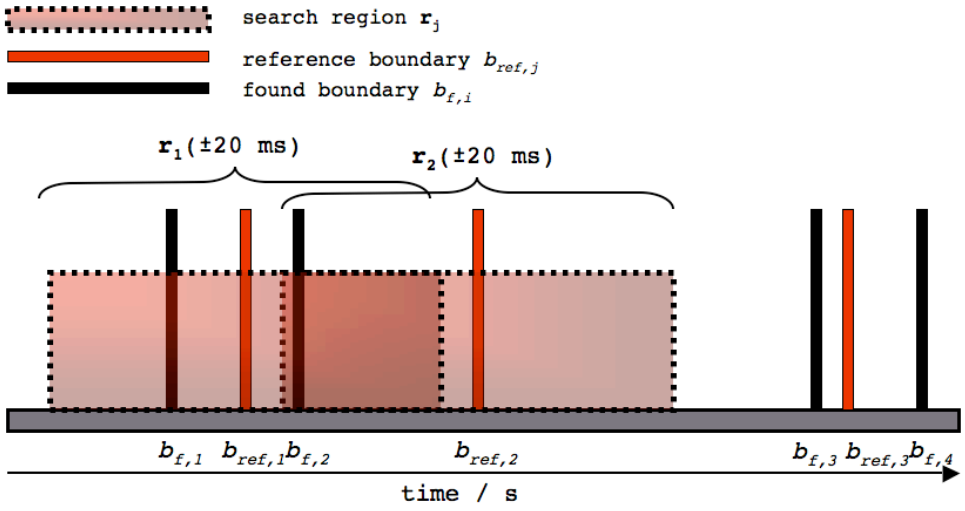

Figure 3.12: Overlapping search regions marked around reference boundaries.

However, there are still possibilities for different interpretations: according to the definition above, one reference boundary is taken at a time and a ±20 ms search region is placed around it. Then it is checked if there are any algorithm boundaries located inside that region. If there are any, it will count as a *hit*. Then a next region is created and searched for boundaries, and if there are any of them inside it, a new *hit is obtained* again, *allowing the use of a same boundary produced by the algorithm to match two different peaks in two subsequent search regions.* As in the case shown in fig 3.12, no deletions would be obtained from either of the search regions $\mathbf{r_1}$ or $\mathbf{r_2}$. The second (tacit) interpretation would be the one where the same algorithm boundary cannot be used in several regions to match reference boundaries. Both of these interpretations lead to different results in an automatic evaluation of the hit rate (which is naturally inversely proportional to the deletion rate).

Other problems also exist: allowing each boundary to be used only once leads to another interesting question - on what basis are the boundaries paired with the reference

boundaries (i.e., banned from further use)? In figure 3.12 we have two search regions marked and one boundary is located inside both regions. If $b_{f,1}$ is paired with $b_{ref,1}$ and $b_{f,2}$ with $b_{ref,2}$, no deletions or insertions occur without violating any of the above interpretations of the evaluation methods. On the other hand, if the evaluation algorithm pairs the closest boundary $b_{f,2}$ to the $b_{ref,1}$, $b_{ref,2}$ causes a miss since the only boundary inside $\mathbf{r_2}$ has already been used.

Before making any further conclusions about what can be actually considered as the "best" method of evaluation, a few more concepts need to be introduced:

A *gross error* is encountered when the segmentation algorithm produces a segment that begins and ends before the corresponding reference segment begins (see fig. 3.13). A *fine error* occurs when there is only a small deviation between the reference and the algorithm boundary, while the alignment of the boundaries still follows the structure of the reference in general (*Kvale,* 1993). *Balducci & Cerrato* (1999) use the terms *position error* and *recognition error* for the same phenomena, but these expressions can be misleading since the segmentation algorithm does not always deal directly with recognition per se.

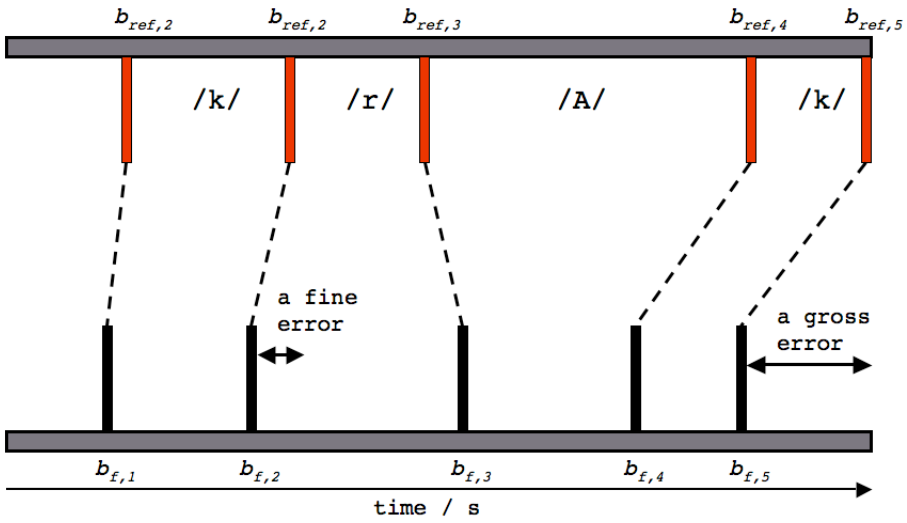

Figure 3.13: Gross and fine errors in segmentation (adapted from *Kvale*, 1993 and *Balducci & Cerrato*, 1999).

Having fine errors (or position errors) does not necessarily mean that the algorithm has found incorrect locations for boundaries. Even gross errors (according to the reference boundaries) can sometimes be considered as correct segmentation, if the original speech signal is examined instead of the reference annotation. This results from the fact that there is no such thing as a universally correct speech segmentation, neither is there only one correct method to perform phonetic transcription. Each boundary in the manual annotation is produced by a human being, which results in an inevitable conditional variability in the segmentation and labeling.

A study concentrating on quality of manual transcription carried out by *Wesenick* and *Kipp* (1996) illustrates this problem. They estimated the congruence of both manual

labeling and segmentation, also comparing them to automatic systems. For labeling, they found that approximately 95 % of manual consonant labels were similar between manual transcriptions performed by different phoneticians. For segmentation, as one might expect, accuracy was highly dependant on the type of transitions that take place between phones: transitions from unvoiced fricatives to unvoiced plosives were most consistently segmented, resulting in an average deviation of 5 ms between different segmenters. However, in nasal-to-nasal transitions, boundaries deviated on average as much as 16 ms. Wesenick and Kipp also made an interesting observation that the boundaries that are difficult for humans to find are also less correctly determined by automatic systems. Similar results regarding the deviation of manual transcription were already observed earlier by *Kvale* (1993), whose work on evaluation of segmentation is often cited when the above methodology or its analogies are used for segmentation evaluation.

On the basis of these findings it can be concluded that using absolute distance metrics does not provide undistorted information about the quality of the algorithm under inspection. If one wishes to evaluate segmentation algorithms using a binary decision (hit/miss) logic, one needs to take into account the noise that originates from reference segmentations. The use of ±20 ms as the maximum allowed deviation from the reference boundaries has been established in the field, possibly because it has been found as a convenient compromise between the variability of annotation and the willingness to keep hit conditions reasonably tight. *Barry* (1991) was possibly the first one who used this ±20 ms value in the distance evaluation while carrying out a labeling study where several automated labeling systems were compared usinng several languages. It is also a paper that several papers cite when segmentation evaluation is discussed. Barry, however, did not specify any justification for this selection.

If we assume that manual annotations deviate with a Gaussian distribution around the "real" segment boundaries, even 40 ms (±20 ms) search regions are not always sufficient to guarantee that the algorithm under evaluation has actually missed the phone boundary if it does not produce a peak in the search region, as can be seen in the example in fig. 3.14.

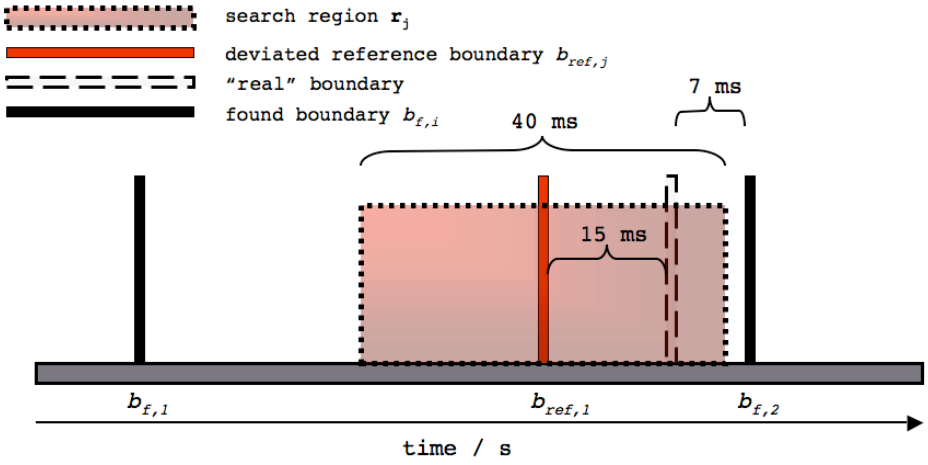

Figure 3.14: A miss of a boundary in the evaluation due to deviation in manual segmentation.

As the figure demonstrates, if the reference boundary is within the limits of typical deviation of manual segmentation (true[9] location shown in dashed lines), and the algorithm has produced a boundary that lies only 7 ms apart from the same location, the situation is still interpreted as a miss. However, while many authors have used larger search regions leading to attenuated effect of variability in the reference (e.g., *Demuynck & Laureys*, 2002), the use of larger hit-zones is easily questioned by, e.g., considering the average length of shorter plosives and the phone rate of normal speech. For example, in TIMIT there are on average 12 phones per second in the speech signals (including the silence before and after the utterances), meaning that there is a segment boundary every 83.3 ms. If each boundary has a 40 ms hit window and it is assumed that they are equally spaced, almost half (48 %) of the signal is covered by search regions. By increasing the window size to 35 ms (84 % coverage), 50 ms (121 % coverage) or even to 70 ms (168 % coverage), it actually becomes a difficult problem to systematically insert boundaries *outside* the search regions. This is the reason why the use of large hit zones does not serve the quality of evaluation very well.

While automated algorithms for segmentation evaluation are extremely helpful in development and testing of new methods, the best understanding of the functionality of the segmentation algorithm can be obtained by manual inspection. Looking and listening to the results at the word and phone level offers deeper insight to the characteristics of the system, allowing the knowledge of the observer regarding the language and context to interact with the data. The additive noise produced by deviations in manual reference annotation and the lack of phonetic comprehension in common automated evaluation methods renders the quantized performance evaluation factors into guidelines that are superficial in nature rather than accurate descriptions of the quality. A search for better and more descriptive automated methods that could form a standard for evaluation would probably be worthwhile, but it is not within the scope of this thesis.

3.2.2 Segmentation evaluation methods used in this study

A well-established DARPA-TIMIT Acoustic-Phonetic Continuous Speech corpus[10] (1993) was used for testing the segmentation algorithm developed within this thesis. The corpus was split up into training and testing data. The training part consists of a total of 4620 signals (spoken by 3260 male and 1360 female speakers) while the testing part has a total of 1680 signals (1120 male, 560 female). Each of the speakers uttered 10 sentences and each sentence was manually annotated using an acoustic-phonetic transcription that included 61 phonetic symbols and segment boundaries. Every sentence

[9] A true location refers here to the mean annotation of several phoneticians.

[10] "*The Texas Instruments/Massachusetts Institute of Technology (TIMIT) corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. TIMIT contains speech from 630 speakers representing 8 major dialect divisions of American English, each speaking 10 phonetically-rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic, and word transcriptions, as well as speech waveform data for each spoken sentence*" (*Garofolo et al.*, 1993)

was spoken in one of the eight major dialects of the United States. The database was recorded using a 16 kHz sampling rate.

In addition, to obtain a better understanding of the language dependency of the algorithm, segmentation accuracy was tested with a small in-house[11] corpus of spoken Finnish. It consists of two male speakers, each speaking 80 and 117 utterances respectively. The database was annotated using the Worldbet machine-readable phonetic alphabet and recorded using a 22050 Hz sampling rate, which was downsampled to 14.7 kHz for the experiments.

Two different methods were used for segmentation evaluation on the basis of the discussion in the last section. *Method I* represents the version where the algorithm moves in the time-domain from the beginning of the speech signal to the end, finding the nearest algorithm produced boundary for each reference boundary, and if they are closer than 20 ms to each other, both boundaries are banned from further use. *Method II* adapts the idea of placing similar ±20 ms search regions around each reference boundary (simultaneously), and calculating the deletion rate, and thereby hit rate, by checking which of the search regions do not contain any algorithm boundaries. In addition to evaluate the quality of the algorithm presented in this thesis, the use of these two different methods also offers a good example to demonstrate how different results can be obtained by using algorithms that cannot be distinguished from each other by the definitions given in the literature.

Method I

One speech signal is evaluated at a time. The testing algorithm takes a boundary $b_{ref,j}$ from a list of boundaries $\boldsymbol{b_{ref}}$ provided by the reference annotation for the corresponding signal, and calculates distances to each of the boundaries in $\boldsymbol{b_f}$ produced by the segmentation algorithm. The closest boundary $b_{f,i}$ found is chosen, and its distance to the boundary $b_{ref,j}$ is inspected. If the distance in time is less than 20 ms, the boundary is considered as matched and $b_{f,i}$ is removed from the $\boldsymbol{b_f}$ to prevent multiple matches. If there is no matching $b_{f,i}$ boundary for $b_{ref,j}$ existing within ±20 ms, it will count as a miss. Each subsequent boundary $b_{ref,j}$ in the annotation is tested for a possible match. The final results are calculated as an arithmetical mean of all results from the evaluated signals.

Method II

One speech signal is evaluated at a time. The testing algorithm takes boundaries $\boldsymbol{b_{ref}}$ from the reference annotation, and places ±20 ms search regions around each boundary $b_{ref,j}$. Boundaries $\boldsymbol{b_f}$ produced by the segmentation algorithm are then mapped onto the same time axis. Each search region is searched for boundaries $b_{f,i}$, and those containing any are counted as hits, while empty regions are counted as misses (deletions). The final results are calculated as an arithmetical mean of all results from the evaluated signals.

[11] Recorded and annotated in the Laboratory of Acoustics and Signal Processing, Helsinki University of Technology.

The quantized performance factors

If boundary matches are denoted with N_{hit}, the total number of boundaries produced by the algorithm with N_f, and the number of boundaries in the reference annotation with N_{ref}, the over-segmentation coefficient and the hit rate are defined as follows (*Petek et al.*, 1996; *Aversano et al.*, 2001):

Over-segmentation percentage:

$$D' = (\frac{N_f}{N_{ref}} - 1) * 100 \tag{3.5}$$

Correct detection percentage:

$$P_c = \frac{N_{hit}}{N_{ref}} * 100 \tag{3.6}$$

In many fields of research, there are two other central criterions for evaluation: precision (3.7) and recall (3.8). In the case of speech segmentation, they can be defined as follows (*Ajmera et al.*, 2004):

$$PRC = \frac{N_{hit}}{N_f} \tag{3.7}$$

$$RCL = \frac{N_{hit}}{N_{ref}} \tag{3.8}$$

Precision describes the likelihood or ratio of how often the algorithm hits a correct boundary when it detects one. Recall is exactly the same as P_c, except it is not multiplied to be a percentage. From these two criterion we can calculate a single scalar (0-1) that can be used to estimate overall segmentation quality, the F-measure (*Ajmera et al.*, 2004):

$$F = \frac{2.0 * PRC * RCL}{PRC + RCL} \tag{3.9}$$

Use of these performance factors should be taken as a guideline in order to understand the effects of changes in the segmentation algorithm or in the material. They do not directly indicate what has been enhanced or what the problems in the process are, and therefore they should not be trusted alone. A direct comparison between different segmentation methods is possible by using these factors, but as already pointed out, the interpretation of hits and misses is not self-evident, nor consistent in literature.

3.2.3 Evaluation methods for features and clustering

During the development of large speech recognition systems it is often also necessary to evaluate the quality of data clustering. In order to improve clustering algorithms and to gain insight into their functionality, there must be some type of metrics that capture the most important and relevant features of the cluster space and represent them in a human understandable form.

Handl and *Knowles* (2007) have outlined that ultimately the quality of a clustering is defined in terms of external expert knowledge. However, they also note that as the clustering methods are usually unsupervised and external knowledge of the clustering process is usually unavailable, one must resort to internal criteria for evaluation. In this thesis the aim was to segment speech into phone-sized units and to cluster these single realizations of phones into corresponding clusters to form an abstract representation of the input speech. To gain insight about the behavior of the clustering algorithm, the manual annotations of corpora are used to describe each produced segment with a corresponding phone distribution. Ideally each segment would consist of only one phonemic class, but because of *1)* some overlap and inaccuracy in segment boundaries between the reference and the algorithm, and *2)* since, e.g., some diphones or triphones may be segmented as one single segment, denoting the phone-distribution as a portion of entire segment duration gives more insight into the statistical properties of the segmentation. Therefore, in the experiments each segment is characterized by one or several phone labels that are averaged with a weight into a cluster's phonetic distribution as the segment becomes merged into a cluster. In this manner it is possible to inspect the distribution of phones in each cluster, one manner in which to evaluate cluster purity.

While knowing the phonetic distribution of each single cluster helps to understand what is actually occurring in the clustering process, the use of single distributions alone is not a very convenient manner to get a good overview of the entire system. As is already known, it is permissible for speech recognition purposes for clusters to be selective only to those features of speech that they are trying to represent on an abstract level. In this case, these abstractions are phones, or more specifically, different realizations of them, and the features are feature vectors from the corresponding segments. Therefore, minimizing the randomness of the phone distribution in a cluster can be compared to increasing the selectivity (and quality) of the clustering. Randomness, or absence of patterns, can be described in terms of *Shannon entropy*. To estimate such entropy of a cluster, we use formula:

$$H = -\sum_{i=1}^{K} p(i) * \log p(i) \qquad (3.10)$$

where K is the number of different phone classes in the reference and $p(i)$ is the distributional probability of the phone class i. In ideal clusters describing single phones, the distributional probability would naturally be zero for each but one phone class, leading to zero entropy. As the phone distribution becomes more complex, the entropy increases. This method of evaluating cluster space purity has been found generally

convenient in the literature (*Duda et al*. 2001). In this thesis, the term *entropy* will always refer to Shannon entropy, or, information entropy.

Entropy can also be used to describe the size distribution of clusters in a cluster space. If we consider N as a number of clusters and $p(i)$ as a relative size of the cluster i (all clusters together summing up to $p = 1$), and by using equation 3.10 again a scalar number between 0 and 1 is obtained that indicates how large the differences are in cluster sizes. The larger the entropy, the more uniform the cluster space is. For robust segmental classification, it may be appropriate to aim at several large clusters that represent different types of realizations of phones with good selectivity, which means that the phoneme distribution entropy has to be minimized and the cluster size entropy has to be maximized, simultaneously.

To further exploit the availability of manual annotation, clustering and segmentation can also be evaluated together in terms of phone recognition: each produced segment is compared to the annotation and the most dominating phone class is chosen. Feature properties are extracted from the segment and are clustered into a pre-existing (so called *trained*) cluster space. The dominating phone class of the cluster, with which the segment merges, is then compared to the dominating phone class of the segment. If these two are of the same class, the phone segment is considered as recognized. Otherwise the segment is classified as misrecognized. The recognition percentage can be calculated over large sets of data to provide statistically reliable information. Some authors also compare several most probable phonetic labels of each cluster (see, e.g., *Zahorian et al.*, 1997), although it can be questioned whether this deals more about recognition or about the possibility to interpret the same data with possibly totally different meanings.
In reality, full exact phone recognition is an extremely difficult problem even for humans and especially in noisy conditions. Many segmentation algorithms, including the one used in this study, are based on the tracking of spectral changes. Without exploiting any context information, this generally causes some phones, e.g., plosives, to be split into a low energetic closure part (occlusion), and a spectrally more significant burst part (release). The closure sections of different phones are difficult to distinguish from each other, as they are only a part of a larger functional unit and consist mainly of silence. Also, slight variations in some nasal and vowel forms are hard to distinguish from each other but they rarely play a central role in the comprehension of speech. If they are processed as single distinct phones, it will lead to higher misrecognition rates in many systems due to the classification of two or more of these variations into the same abstract category. Therefore, many research papers using TIMIT annotation for recognition also use the so called reduced phone-class set (*Antal*, 2004; *Zahorian et. al.*, 1997), as will also be used in the experiments in this study (see table 3.1).

It should be noted though, that TIMIT is a corpus spoken in American English, and this type of classification of single phones into larger sets is predominately based on linguistic properties of spoken English. In some other languages, some of the phones classified into the same classes with this reduced set might have important meaning distinguishing properties between each other. Therefore this kind of reduced phone set cannot be

directly applied to speech material existing in other languages, even while they would be also annotated in a similar fashion.

Table 3.1: Reduced TIMIT phone set (according *Antal*, 2004).

Category	Group	Category	Group	Category	Group
Vowel	ah, ax, axh	*Semivowel*	el, l	*Fricative*	f
Vowel	iy	*Semivowel*	r	*Fricative*	th
Vowel	ih, ix	*Semivowel*	w	*Fricative*	v
Vowel	eh	*Semivowel*	y	*Fricative*	dh
Vowel	ey	*Semivowel*	hh, hv	*Stop*	b
Vowel	ae	*Nasal*	m, em	*Stop*	d
Vowel	aa, ao	*Nasal*	n, en, nx	*Stop*	g
Vowel	aw	*Nasal*	ng, eng	*Stop*	p
Vowel	ay	*Affricate*	jh	*Stop*	t
Vowel	oy	*Affricate*	ch	*Stop*	k
Vowel	ow	*Fricative*	s	*Stop*	dx
Vowel	uw, ux	*Fricative*	sh, sz	*Closure*	epi, q, bcl, dcl,
Vowel	axr, er	*Fricative*	z		gcl, kcl, pcl,
					tcl, pau, #h

4. Experiments and findings

In order to develop and test the methods involved in the bottom-up process, several experiments were conducted in the MATLAB-programming environment. As a necessary front-end operation in the architecture presented in this thesis, the segmentation algorithm was tested first. After the segment quality evaluation, different approaches to clustering will be evaluated and discussed. To understand the implications of the different feature extraction settings, the feature extraction evaluation was tightly coupled with the clustering evaluation and its evaluation methodology.

4.1 Experiments with segmentation

The segmentation algorithm was tested for clean speech with English and Finnish material. Noise robustness was also evaluated, leading to interesting results concerning the hit rates with high over-segmentation rates. These results, with a brief analysis of the underlying statistics, will be the main points covered in this section.

4.1.1 Segmentation of English material

The segmentation algorithm was mainly tested with the two different methods (*Method I* and *Method II*) described in the previous chapter. The aim was to get a good understanding of the overall performance of the algorithm that could be compared to the other results reported in blind segmentation literature, and to determine the general effects of different parameters to the segmentation results.

The first results, presented in table 4.1, contain the evaluation of a basic segmentation of the TIMIT test section with default settings. Full test sets, N_{fem} = 560 for female and N_{male} = 1120 for male speakers, were used, containing utterances from a total of 168 different speakers. As can be seen from the results, the algorithm is not very susceptible to gender specific differences. Also, the difference between the two evaluation methods can be easily seen: *Method II*, in which all of the search regions are searched for peaks simultaneously, results in an approximately five percent better hit rate than *Method I,* while the actual segment boundaries are located in the exactly same positions in both

cases. The hit-rate is denoted with P_c, over-segmentation with D', precision with PRC, recall with RCL, and the F-measure with F.

Table 4.1: Segmentation results for the TIMIT test-section with negligible over-segmentation.

Data	Method	P_c	D'	PRC	RCL	F
test/female	I	75.59	0.87	0.75	0.76	0.76
test/female	II	80.91	0.87	0.80	0.81	0.81
test/male	I	75.10	0.43	0.75	0.75	0.75
test/male	II	80.26	0.43	0.80	0.80	0.80

By accepting higher values of over-segmentation, higher hit rates can be obtained. The most efficient way to do this with the algorithm is to adjust the length of the Minmax-filter and the probability threshold p_{min} of the peak detector. Table 3.3 shows the results for the train/female set with two different amounts of over-segmentation.

Table 4.2: Segmentation results for the TIMIT test/female set with two different amounts of over-segmentation.

Data	Method	P_c	D'	PRC	RCL	F
test/female	I	79.07	17.92	0.67	0.79	0.73
test/female	II	84.37	17.92	0.72	0.84	0.77
test/female	I	83.79	42.10	0.59	0.84	0.69
test/female	II	88.77	42.10	0.62	0.89	0.73

These results obtained are well in line with the other results reported in literature regarding blind segmentation algorithms. For example, *Aversano et al.* (2001) reported a hit rate of $P_c = 73.58$ % with an over-segmentation value $D' = 0$ %, or by allowing excess over-segmentation, they gained $P_c = 90$ % and $D' = 63$ %. Recently, *Estevan et al.* (2007) have reported a hit rate of $P_c = 76.0$ % with non-existent over-segmentation $D' = 0$ % while at $P_c = 90.3$ %, $D' = 75.5$ %.

4.1.2 Segmentation of Finnish material

As for the Finnish database, both speaker's speech was automatically segmented independently to gain insight to both a) single speaker dependency, and b) the difference between rather swiftly spoken English material and very carefully articulated Finnish speech. The results are shown in table 3.4.

Table 4.3: Segmentation results for the Finnish corpus, two speakers.

Data	Method	P_c	D'	PRC	RCL	F
Speaker-1	I	85.86	41.53	0.61	0.86	0.71
Speaker-1	II	86.74	41.53	0.61	0.87	0.72
Speaker-2	I	87.80	45.30	0.60	0.88	0.72
Speaker-2	II	89.24	45.30	0.61	0.89	0.73

As can be seen, the segmentation algorithm finds more boundaries in Finnish speech than in TIMIT material. This is an expected result even if general phonetic differences between English and Finnish are ignored, since the Finnish material is spoken more slowly and each syllable is being stressed with greater care. It is also noteworthy, that in addition to gender invariability, the results for two totally different sounding speakers are also rather similar in terms of F-values.

The large difference in over-segmentation between TIMIT and Finnish corpora is mainly due to differences in annotation: while TIMIT and the Finnish corpora both used an acoustic-phonetic labeling scheme, only TIMIT included differentiated closures and bursts for plosives while Finnish data considered plosives as one single unit. As the algorithm searches for possible segment boundaries by tracing changes in the spectrum, it will (in optimal situation) create three boundaries for each plosive instead of the two that can be found in the Finnish corpus' annotation.

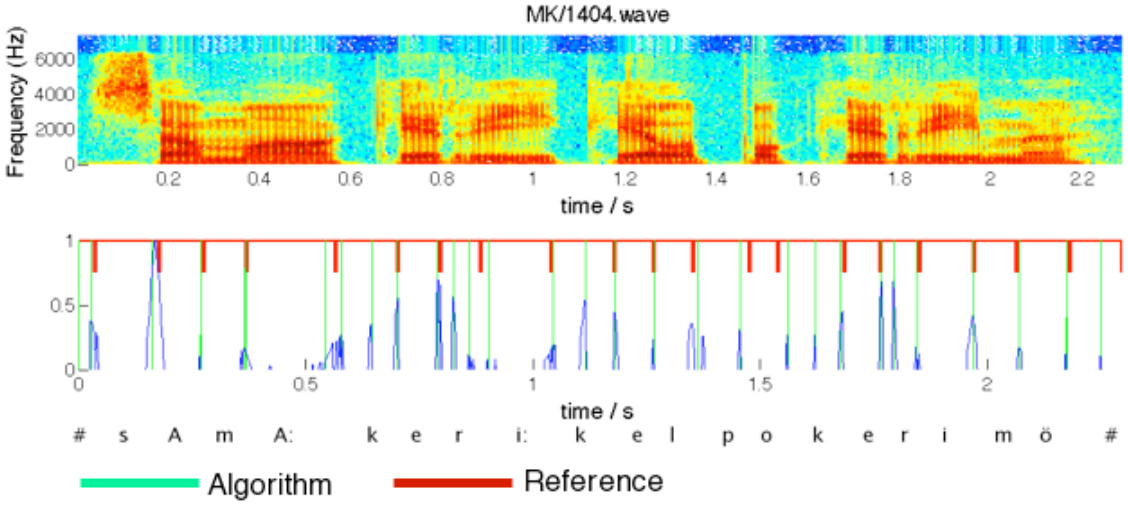

Figure: 4.1: Example of segmentation results for a Finnish utterance, *"Samaa kerii kelpo kerimö",* with a spectrogram and the segmentation output with reference marked.

In figure 4.1 the utterance *"Samaa kerii kelpo kerimö"* is represented as a spectrogram paired with the output of the Minmax-filter (blue peaks), output boundaries of the segmentation algorithm (green lines) and the annotation reference (red lines). Algorithmically generated boundaries are very well matched with the spectral changes, and in most of the cases also with the annotation reference. However, as one can observe, the algorithm has produced extra boundaries for the burst of plosive [k] at t_1 = 0.6 s, t_2 = 1.1 s and t_3 = 1.6 s. Also, a general characteristic difference between the algorithm and the annotations can be found at the endings of the speech signals: here the last vowel [ö] is denoted to end at approximately $t_{ö}$ = 2.18 s, while the spectrum of the breathy ending keeps fading away for few hundred milliseconds longer. As the algorithm reacts most prominently to the point where there is a discontinuity point in the spectrum (i.e., the signal changes from a correlating formant structure to a silence), it places two separate boundaries at the end of the utterance: one at the sudden change of the structure, and one

where the spectrum of the breath finally fades to a non-existent level. This effect is observed with both English and Finnish data.

4.1.3 Parameter dependency

There are many parameters in the segmentation algorithm presented in this thesis that can be adjusted to produce alternative results for speech segmentation. Many of these are complementary, and in some cases the correct values depend directly on other parameters. The most central of these settings have been gathered in table 4.4 shown below. The *range* value describes the usual or meaningful value range for each parameter, and the set of default values is defined such that using all of the default values at the same time will result in relatively stable and invariant results with all tested speech material.

Table 4.4: Central parameters of the segmentation algorithm.

Parameter	Explanation	range	default
wl	FFT window length	n/a	96
ws	FFT window step	8–*wl*	32
d_1	2D-integrator box size	2–30	10
d_2	2D-integrator triangle size	2–30	10
alpha	Tanh[x]-compression coefficient	0–1	0.45
p_{min}	Threshold for peak selection	0–1	0.06
n_{mm}	Length of the Minmax-filter	5–60	40
t_d	Distance of the peak masking effect	0–100	0.025

During all of the experiments, including the parameter tests, the FFT window length *wl* was set to a constant of 96 samples (6 ms for a 16 kHz sampling rate). As the purpose was to perform almost pitch-synchronous FFT-analysis, the window size is supposed to be approximately the same length as of one pitch period. The 6 ms window generally satisfies this condition for both male and female speakers. It should also be noted that while the FFT is usually computed with window lengths that are powers of two, no zero-padding was used since MATLAB uses a combination of several numerically sophisticated techniques to perform the calculations, and the computational speed was found to be even faster for a 96 point window compared to a 128 point transformation.

During the development of the algorithm, it turned out that the length n_{mm} of the Minmax-filter, threshold p_{min} and masking distance t_d of the final peak selection were the most dominating parameters for the hit rate P_c and over-segmentation rate D'. As for the n_{mm}, the value is mainly a tradeoff between over-segmentation and hit-rate, where approximately $n_{mm} = 40$ was used in most of the tests to produce approximately $D' = 0$ % over-segmentation.

For the purpose of testing and demonstration, a small number ($N = 100$) samples from TIMIT test/female was evaluated with different parameters to describe the behavior of the algorithm as a function of parameter settings. All results were evaluated by using

Method I. It is important to note that all of the other parameters were kept constant during the evaluation of the chosen single parameter, and these other values were set to the default values. The effects of the length of Minmax-filter are listed in table 4.5.

Table 4.5: The effect of length n_{mm} of the Minmax-filter on segmentation results.

P_c	D'	F	n_{mm}	d_1	d_2	ws	p_{min}	t_d
70.87	-4.54	0.73	50	10	10	32	0.06	0.025
74.59	3.99	0.73	40	10	10	32	0.06	0.025
77.41	12.68	0.73	30	10	10	32	0.06	0.025
78.78	18.35	0.72	20	10	10	32	0.06	0.025
78.73	18.87	0.72	10	10	10	32	0.06	0.025

As can be observed, the length n_{mm} controls the tradeoff between D' and P_c, but the F-value is not greatly affected by the changes. Table 4.6 shows the effects of adjusting the peak detection threshold p_{min}.

Table 4.6: The effects of threshold for peak selection p_{min} on segmentation results.

P_c	D'	F	n_{mm}	d_1	d_2	ws	p_{min}	t_d
79.61	27.36	0.70	40	10	10	32	0.02	0.025
77.63	17.05	0.72	40	10	10	32	0.04	0.025
74.59	3.99	0.73	40	10	10	32	0.06	0.025
71.19	-6.81	0.74	40	10	10	32	0.08	0.025
68.15	-14.98	0.74	40	10	10	32	0.10	0.025

The peak selection threshold value p_{min} has a more dramatic effect on the F-value. This is an expected result, since it resembles the probability threshold for boundary detection: the more probable peaks that are chosen, the better the precision that is obtained. However, when using higher values of p_{min} the algorithm starts to miss less probable (in terms of the algorithm), but still existing, phone-boundaries. As the aim of segmentation is to find as many boundaries as possible, higher F-values at the cost of very low hit rates are not eligible.

Table 4.7: The effect of masking distance t_d on segmentation results.

P_c	D'	F	n_{mm}	d_1	d_2	ws	p_{min}	t_d
78.91	25.11	0.70	40	10	10	32	0.06	0.005
78.91	25.11	0.70	40	10	10	32	0.06	0.015
74.59	3.99	0.73	40	10	10	32	0.06	0.025
68.75	-8.31	0.72	40	10	10	32	0.06	0.035
65.60	-11.83	0.70	40	10	10	32	0.06	0.045

Table 4.7 indicates the results for masking distance t_d. A sweet spot can be found in the proximity of t_d = 25 ms. This is a reasonable result, since the rate of articulation in normal speech rarely exceeds four phones per 100 ms. There are still, e.g., some very quick plosives that can have bursts shorter than 20 ms, resulting in decreased P_c with longer masking distances than burst durations. On the other hand, by using values of tens of milliseconds, segmenting longer bursts into several small segments is avoided, since

the cross-correlation of the spectral coefficients may vary considerably within such variable transitions.

Table 4.8: The effect of window step size *ws* on segmentation results.

P_c	D'	F	n_{mm}	d_1	d_2	ws	p_{min}	t_d
81.7	43.21	0.67	40	10	10	16	0.06	0.025
77.88	17.2	0.72	40	10	10	24	0.06	0.025
74.59	3.99	0.73	40	10	10	32	0.06	0.025
71.54	-3.74	0.73	40	10	10	40	0.06	0.025
68.17	-10.96	0.72	40	10	10	48	0.06	0.025

The step size of the FFT window has two important properties: it defines how often and how many of the windows will be applied to the waveform pitch-synchronously, and it also has a significant effect on computational time. A sweet spot in terms of the *F*-value can again be found at the proximity of the default value of 32 samples, that corresponds to 2 ms steps with the 16 kHz sampling rate used in the TIMIT tests, or 2.2 ms with the 14700 Hz Finnish database.

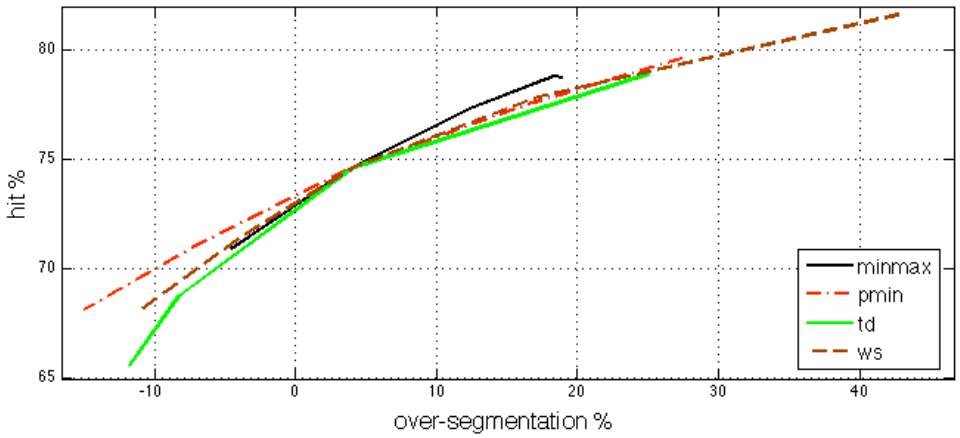

Figure 4.2: Effects of different parameter values on segmentation results tested independently of each other.

Summarizing the parameter dependencies, most parameters control the tradeoff between over-segmentation and hit rate, and no parameter alone has an unquestionable effect of improving the results (see fig 4.2). Also, since many of the parameters are complementary, there are many possible combinations that achieve very similar results. Each value choice for a parameter limits the maximum hit rate P_c by some amount in order to keep the over-segmentation at reasonable levels. Therefore, in theory it could be possible to achieve much higher hit-rates by allowing *D'* to grow to very high values (see table 4.2). However, then the possibility of hitting the correct search regions, just by randomly inserting the boundaries, starts to dominate the process as will be seen in the next section.

4.1.4 Error analysis and noise robustness

The noise robustness of the algorithm was tested by adding white Gaussian noise to the TIMIT speech signals before evaluating the segmentation quality. The signal-to-noise ratio (SNR) was calculated by taking the mean power of the original signal over the entire signal duration, and then comparing it to the mean power of the white noise signal that was added to the original signal.

Table 4.9: Hit rates with added white noise.

P_c	D'	F	SNR
73.69	-2.95	0.75	baseline
76.36	11.69	0.72	38.17
78.43	16.60	0.73	27.71
79.77	19.97	0.73	21.68
76.84	31.92	0.66	7.7
73.91	35.91	0.63	1.7
70.78	37.32	0.6	-2.9
64.47	39.44	0.54	-∞

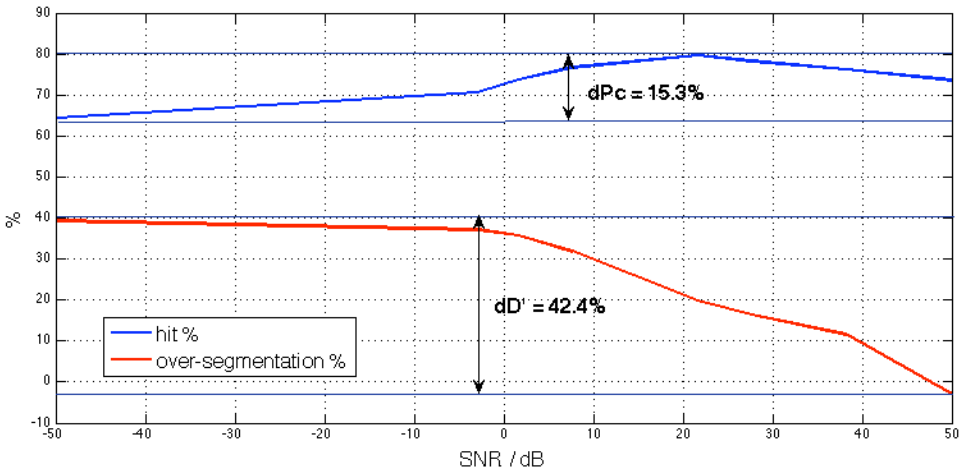

Figure 4.3: Hit rate and over-segmentation as a function of SNR.

Interestingly, the hit rate first started to increase as non-correlating white noise was added to the original speech (fig. 4.3). Also, at very poor SNR-levels the hit rate P_c seemed to converge asymptotically to a specific value, as did the over-segmentation D'. By using pure noise (SNR = -∞) as the only input, almost 65 % of the reference boundaries were still found correctly with an approximately 40 % over-segmentation level. What happens to the performance factors if the entire speech waveform is replaced with noise, and the segmentation results are compared to the reference boundaries? By adjusting the length of the Minmax-filter, the number of insertions for each time unit also becomes adjusted. Figure 4.4 illustrates the consequences of this procedure, when the hit rates and over-segmentation percentages are calculated with different filter length n_{mm}-values.

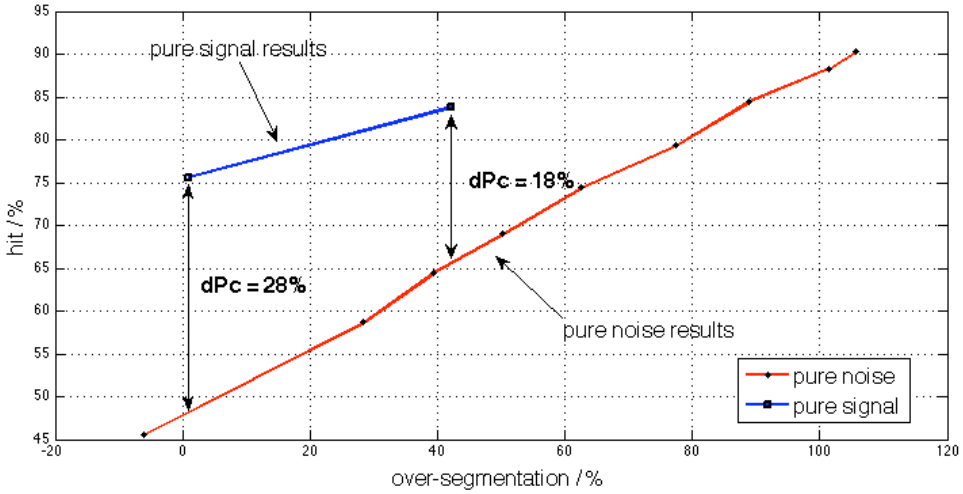

Figure 4.4: Results of segmentation with pure white noise signal.

By feeding the algorithm with pure noise, boundaries start to become detected by random hits to the search regions. As we saw in the chapter dealing with evaluation methods, the search regions around reference boundaries cover a relatively large portion of the timeline. As a result, the chance to hit each reference region by inserting boundaries randomly starts to grow as we allow larger numbers for over-segmentation. Moreover, the nonlinear Minmax-filter and peak masking operation of the algorithm issue special constraints to the locations of the random insertions. This leads to a better P_c/D' ratio than with a pure Gaussian process. One way to interpret this result would be to consider the non-linear filtering used in the algorithm as a kind of modeling of speech prosody and temporal contours. In any case, and with or without any filtering processes, this observation leads to a conclusion that the segmentation results that are reported with relatively high over-segmentation values in the literature say very little about the true nature of the algorithm. The noise-segmentation results can be used to define a zero-level segmentation quality, that can be achieved with systematic insertion of boundaries without any true knowledge of the underlying speech signal.

Figure 4.4 also shows the results for segmentation of pure signals with two different over-segmentation values. The distance dP_c between the noise-segmentation and the pure-signal results starts to decrease rapidly when greater over-segmentation is allowed. This might infer that the process of random insertions may start to dominate the placing of prospective boundaries instead of truly discovering otherwise missed boundaries. There are papers in the literature (e.g., *Kvale*, 1993; *Scharenborg et al.*, 2007) that report extremely high over-segmentation rates in their results, which should not be the case for methodological quality evaluation in the light of these findings.

To gain a better understanding of the nature of the segmentation errors in the algorithm, the full train/male section of TIMIT was segmented to estimate segment boundary deviation from the reference. This is illustrated in fig. 4.5 below. As the previous results predict, the majority of the boundaries are located closer than 20 ms from the reference (mean deviation σ = 13 ms). Also, the mean of the distribution is symmetrically at zero.

However, nearly half of the boundaries are located outside the search region if the permitted deviation is changed from 20 ms to 10 ms. This can be verified by re-calculating the hit rate by using this stricter threshold, which results in about a 55 % hit rate. As long as it can be presumed that the algorithm is mainly reacting to the "real" phonetic segment boundaries, this supports the convention of the 20 ms deviation allowance found in literature.

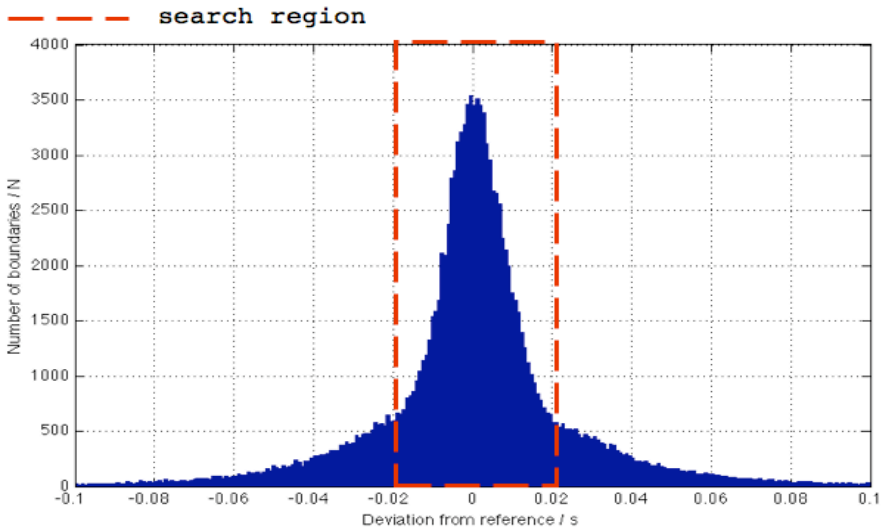

Figure 4.5: Algorithm boundary deviation from the reference, full train/male set.

By hypothesizing that there are several normally distributed processes underlying the ultimate segmentation results, one can consider the following: insertions can be thought of as errors originating from a random process, often for reasons discussed above (e.g., the endings of the signals). Deviation from the reference consists of two normally distributed interactive processes, deviation due to the manual segmentation performed by humans, and the deviation of the algorithm boundaries from these references when the boundary is correctly found. Combining these normally distributed processes would result in yet another distribution, where we should actually see two superpositional normal distributions. Figure 4.6 supports this hypothesis.

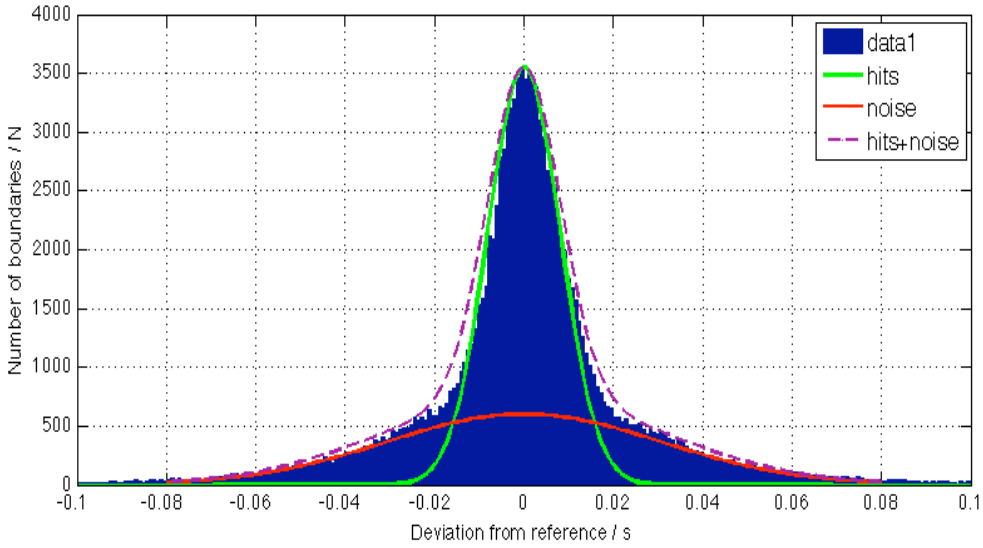

Figure 4.6: Segmentation results as a combination of two Gaussian processes.

The *hits* distribution is the unnormalized normal distribution fit to the center part of the data. It has variance $\sigma_n^2 = 0.09^2$ with mean $\mu_n = 0$. The *noise* distribution is the unnormalized normal distribution that has variance $\sigma_n^2 = 0.175^2$ with a mean $\mu_n = 0$. If the segment deviation distribution is denoted with $P_{seg}(\theta)$ where θ is the deviation from the reference, eq. 4.1 is obtained.

$$P_{seg}(\theta) \sim [N(\mu_n,\sigma_n^2) + N(\mu_h,\sigma_h^2)] \qquad (4.1)$$

$$= P_{seg}(\theta) \sim [N(0,0.175^2) + N(0,0.09^2)] \qquad (4.2)$$

Note that the since the distributions are not normalized, their mixture is not leading to a normal distribution when combined. It should be also noted that the *hits* distribution is almost totally inside the predefined ±20 ms search region.

To further explore the nature of the *noise* distribution above, a normal distribution was fit into a histogram that contains deviation from the reference when only white noise (see beginning of this section for the explanation) was used to replace the speech signal input for the algorithm. By using the same $\sigma_n^2 = 0.175^2$ with mean at zero as above and scaling the distribution to match the numbers of boundaries in the noisy data, very high correspondence between the data and the model $\sim N(0,0.175^2)$ is obtained (fig. 4.7).

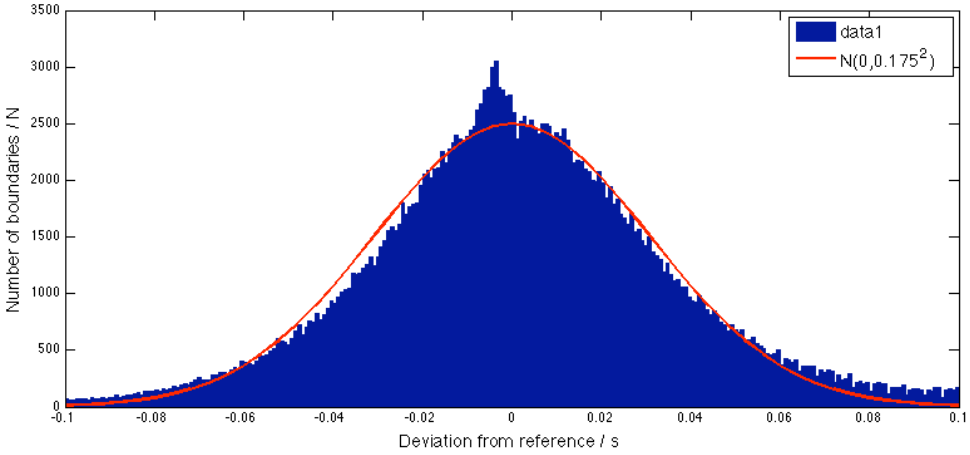

Figure 4.7: Normal distribution model of the noise distribution process fit to the segmentation results of white noise.

Based on these findings it can be hypothesized that the output of the segmentation algorithm is a product of (at least) two different interfering processes. Most of the boundaries are found in similar locations as in the reference by tracking the spectral changes of the speech signal. These boundaries reside normally distributed in the near proximity of the reference boundaries with a mean deviation of less than 10 ms. The deviation of this distribution is most probably defined by the variance in the manual annotation, more specifically by the way the phoneticians use spectral information in the segmentation, and by the small fluctuation caused by limited accuracy of the algorithm in the time domain. Another group, or distribution, of segment boundaries is produced by a normally distributed noise process. Based on experiences with manual inspection, they may occur due to, e.g., insertions in long fricatives and during silence. Also, in the endings of the signals the fading of the spectrum causes deviations between the reference and the algorithm.

The nature of the algorithm is such that the Minmax-filter defines the length of the window where peaks will become inserted, as it was shown in the case of segmenting noise. The peak height is defined locally as the difference between maximum and minimum values of the 2D-filtered signal. This may result in relatively small peaks for some of the real transitions when there are several transitions close to each other, and to relatively large peaks inside a long continuous phone if there are no contrasting transitions nearby. The scaling of the probability classifications in different situations could be one topic of interest in order to improve the algorithm. Also, the properties of the non-linear Minmax-filter should be investigated further.

The use of Gaussian mixtures for evaluating the segmentation algorithm could be a method that takes into account the “random” hits that are produced by excess over-segmentation. Comparison of the size of the *hits* distribution to the *noise* distribution could depict how large a portion of the segments are actually found by using spectral cues, and how much is a result of “good luck”. This does not, however, remove the problems of variance in manual annotation or the fact that there is not a single correct way to perform speech segmentation.

4.1.5 Conclusions of the segmentation experiments

The results obtained by the current version of the algorithm are comparable to those found in current literature (*Aversano et al*., 2001; *Scharenborg et. al*, 2007; *Estevan et al*., 2007), although the method of evaluation is not strictly defined in all papers. The *Method I* evaluation used in this thesis is the strictest possible version of the search region evaluation, and the results obtained with it are extremely close to other best blind algorithms. It is striking that while all authors are using different approaches to create blind acoustic-phonetic segmentation of speech, the results are very similar (also noted in *Estevan et al*.'s paper), and are not able to achieve hit rates higher than 80 %. This could be strong evidence for the necessity of some sort of intelligent top-down feedback that could be used to improve the segmentation accuracy and reliability. However, it should also be kept in mind that the reference comparison is not the ultimate goal of speech segmentation. In the end, the purpose of the segmentation algorithm depends on the entire speech processing system, in which it is implemented, and the most important evaluation method would be then to observe the functionality of the complete system.

This work has also brought out the importance of understanding the nature and purpose of evaluation methods. Most papers in the speech processing literature concerning new segmentation technologies or comparisons between different methods report their results by using a handful of the performance factors presented in this thesis. While this may be indicative towards the overall performance of the systems, one cannot do any extensive and reliable comparisons between different methods from different authors unless a strict standard for evaluation is defined. The use of manual segmentation as a reference may also be unavoidable, but its characteristics should be observed and taken in account when working with the evaluation of segmentation.

4.2 Experiments with clustering and feature extraction

The clustering method presented in this thesis is specially tailored for incremental processing with no practical upper limit for the amount of data. However, due to practical time and resource constraints, limited data sets were used for most of testing and evaluation. The purpose of these experiments is to gain a good basic understanding of the behavior of the algorithm and of the effects of different special means to control the clustering process. Only features extracted from the first half of the segments were used in these experiments, as they are hypothesized to describe the segments better in terms of single static units, and are therefore sufficient for testing the basic functionality of the clustering methodologies used. The latter half of the segment, which is considered to be more context-dependent due to co-articulation, is ignored in this experimental framework and its use is left for further research.

4.2.1 Clustering experiments using the Single-space Method

Before starting the clustering evaluation with the aid of entropy metrics, the base level entropy of the segments was estimated with N = 100 utterances (4066 segments). The mean entropy over every segment in the material (segmentation with 0 % over-segmentation and 74 % hit-rate) was H_c = 0.080, which should be considered as the theoretical maximum (or entropic minimum) for the selectivity of the clusters with respective material.

The first actual clustering experiment was performed with female spoken material from TIMIT (N = 560 sentences consisting of a total of 21821 segments) using the *Single-space Method*. The correlation radius threshold t_m was adjusted to demonstrate the selectivity of the clusters as a function of segmental correlation. Results are shown in table 4.10. H_c is the mean entropy of the cluster phone-distribution and H_s is the entropy of the cluster size distribution. N_{tot} is the total number of clusters and N_{ave} is average number of segments in each cluster. Small clusters (N < 3 segments) are merged into larger clusters and clusters with higher correlation than t_{mm} > 0.9 are merged to each other as a batch process during and at the end of clustering. No integrative cluster centroids or automatically adjusting cluster correlation radii were used. Division to voiced and unvoiced sub-spaces did not have any noticeable effect on the results, so the decision to run tests with a division to voiced and unvoiced clusters was arbitrary.

Table 4.10: Clustering results for the *Single-space Method*.

t_m	N_{tot}	N_{ave}	H_c	H_s
0	13	1379	0.713	0.889
0.2	36	606	0.626	0.814
0.4	118	185	0.514	0.804
0.6	355	61	0.458	0.798
0.8	1057	21	0.399	0.826

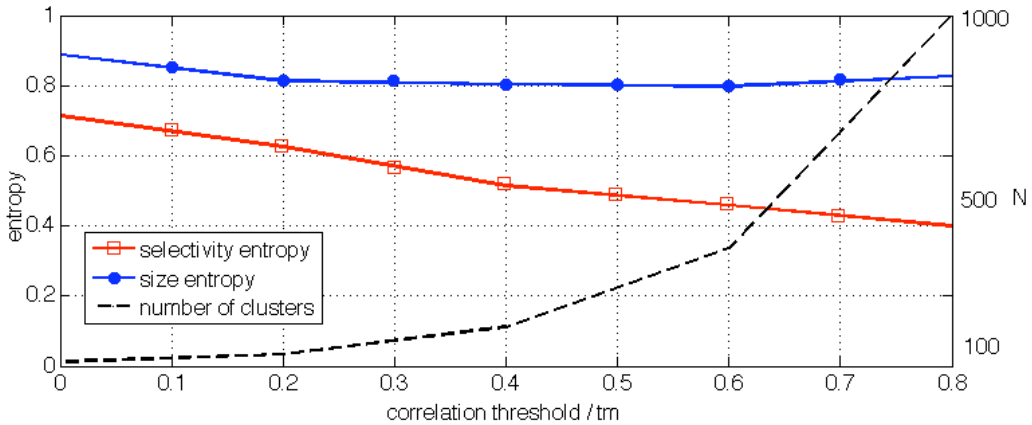

Figure 4.8: Entropies H_c and H_s and the total number of clusters as a function of correlation threshold t_m.

As expected, the entropy H_c of the cluster selectivity starts to decrease as the correlation threshold is increased (fig 4.8). The cluster size entropy H_s on the other hand seems to stay relatively stable, decreasing only slightly when the number of clusters grows. This

can be interpreted as a good property of the algorithm, as the balance of comparative cluster sizes does not seem to be greatly affected by a change in the distance threshold.

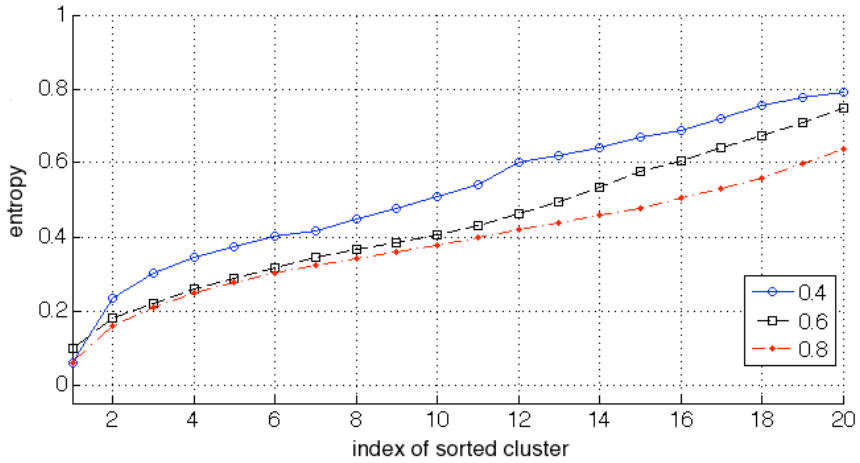

Figure 4.9: Entropy H_c of cluster phone distribution with different correlation thresholds t_m, sorted and normalized indices as the x-axis.

Figure 4.9 illustrates the differences in phone distributions between clusters with several t_m values. Clusters are sorted by their distribution entropy, and the indices are normalized to have a total number of 20 data points. This is because in practice there are order of magnitude size differences in the number of clusters with different thresholds t_m. As can be seen from the figure, there are some clusters that have very low entropy, but most of the clusters fall between H_c = 0.2 and H_c = 0.7 with nearly linear distribution. The increase in threshold decreases the average entropy, but the difference between t_m = 0.6 and t_m = 0.8 is very small in half of the clusters. Also, in practice there are always a few very large clusters with very high entropy, but they do not show up in the figure due to the normalization. Manual inspection reveals that these large clusters often contain not only several variations of similar vowels, but also trills (/r/), laterals (e.g., /l/) and approximants (/w/).

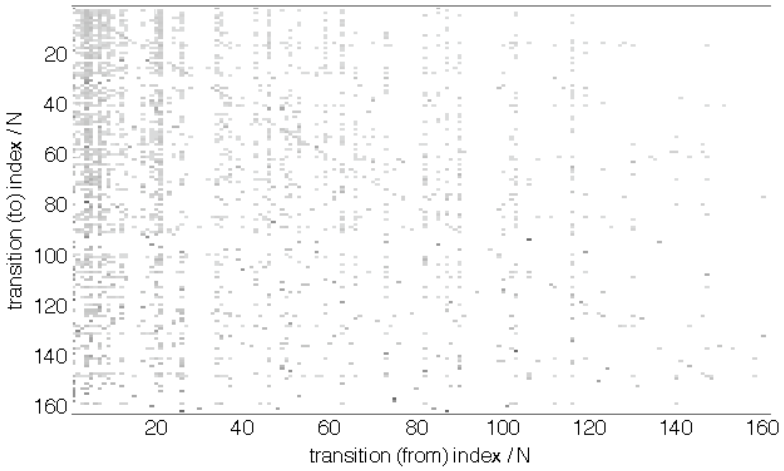

Figure 4.10: Segment transitions as cluster indices. Transitions take place from columns to rows. Threshold t_m = 0.5.

Transitions from segment to segment were also tracked and can be described in terms of the cluster indices of the segments (fig. 4.10). The matrix shows all transitions from all single clusters to every other cluster in the space. The fill percent of the transition matrix, that is, the number of non-zero elements in the matrix containing all possible transitions from one cluster to another, was 17.26 % in the case for figure 4.10. As can be seen, there are very little transitions from clusters with a large index to clusters with a small index. This is an understandable phenomenon when dealing with a limited amount of utterances, as the clusters with a small index are created first while the latter ones are new representations of the subsequent utterances that do not fit well to any pre-existing category. The entropy for the average distribution of transition probabilities over all clusters was also calculated, producing H_t = 0.4805. This means that from an average cluster there are transitions to several other clusters. The diagonal of the matrix contains only a fraction of all transitions, which indicates that there are only very few boundary insertions within similar spectral structures in segmentation, i.e., over-segmentation due to the splitting of phones.

Phone recognition was also briefly tested by clustering the entire TIMIT female test section data with correlation thresholds t_m = 0.4 and t_m = 0.8 and then using N = 50 first utterances of the test set for comparison. A reduced TIMIT phone set was used. Note that the amount of clusters was now only a fraction of the amount of segments in the material, the average cluster size being approximately 100 and 21 segments per cluster, none of the clusters containing less than three segments. For t_m = 0.5, 38 % of phones were recognized correctly, and for t_m = 0.8 the value was 39 %. Interestingly, taking in account three instead of just the first dominating label (see section about evaluation methods for description), the recognition rates were 64 % for 0.5, and 58 % for 0.8. To test and compare recognition with different speakers and different material, utterances that were not used in the clustering material were also tested. The recognition result for t_m = 0.5 clustering was now 34 % with the most dominating label and 61 % when using the three most dominating labels.

The effect of merging small clusters into larger ones and nearby clusters to each other was also tested in terms of phone recognition. 500 signals from the TIMIT train/female material were segmented and clustered incrementally (t_m = 0.8) without any post-processing in the cluster space. 50 utterances from the TIMIT test/female set were used for recognition. In the non-merged case, the recognition result was 30 %, while there were a total of 3712 clusters. Then the clusters with two or less segments were merged to the nearest large cluster and all clusters closer than a correlation radius of t_{merg}= 0.9 to each other were merged together, reducing the total number of clusters to 923. The recognition percent was identical, 30 %, to the one before merging. This seems to indicate that the extremely small clusters would not be statistically relevant from the perspective of the entire classification process.

It should also be noted that the computational complexity of the *Single-space* algorithm starts to increase radically as the number of clusters becomes large (i.e., merge criterion is very tight). The nearest cluster is always searched for from the entire space, indicating that the distance from the segment has to be calculated to every other cluster separately.

This is one of the disadvantages of using a single large space. However, the implementation used in the experiments was not optimized for speed, which may have resulted in some excess computation time. Methods for faster search by, e.g., intelligent cluster indexing through partitioning might be possible without any noticeable degradation in the results.

4.2.2 Effects of centroid integration and adaptive thresholds in the Single-space Method

To estimate how cluster movement in the cluster space affects the clustering results, an alternative method for upgrading the cluster centroids was tested. Instead of averaging every new vector to the centroid with weight $1/N$, where N is the number of segments already in the cluster, new segments were averaged with a constant 5 % weight after the corresponding cluster had already received $N = 20$ or more segments. This prevents the cluster from converging into a specific spot in the space, a location that is mostly dominated by the first few arriving segments. The question with the integrative method is then how should the changes in the phonetic distribution caused by incoming segments be defined? In previous experiments, the $1/N$ weight was given to each segment's features, and similarly each phone distribution of a segment was taken in account with a $1/N$ weight in the cluster's phone distribution. Using a 5 % constant weight for new distributions would mean that previous utterances have less weight for the final distribution during the evaluation. To maintain the results comparable with the other experiments, the phonetic distribution was still updated with a $1/N$ weight for each segment, while the centroid was updated with the constant 5 % weight after a number of 20 segments was reached. One should keep in mind that in this experimental setting the cluster location does not match the distribution of phonetic content in segments directly. 200 utterances were used to test the difference in entropy with integrative centroids. The results have been gathered into table 4.11.

Table 4.11: Clustering results without and with integrative centroids.

t_m	$H_{c,n}$	$H_{c,i}$	$H_{s,n}$	$H_{s,i}$	$N_{ave,i}$	$N_{ave,n}$
0.5	0.478	0.483	0.784	0.792	73	84
0.6	0.438	0.432	0.795	0.788	45	43

$H_{c,n}$ is the entropy of the phone distributions without integration while $H_{c,i}$ includes integration. Similarly, $H_{s,n}$ is the entropy of the cluster size distribution without integration, and $H_{s,I}$ is with integration. Adding new segments with a 5 % weight instead of $1/N$ increases the entropy slightly with $t_m = 0.5$, but decreases it with $t_m = 0.6$. The average number of segments in the clusters is nearly constant with both methods. Overall, integration is not causing any significant changes in the results.

Another subject of interest was the adaptive correlation threshold: the more segments that end up in a cluster, the more selective the cluster becomes. This feature was tested in combination with integrative clustering. The threshold was defined with the following formulae:

$$t_{m,i} = t_{m,0} + a_1 N_i, \qquad N_i < N_{\max}$$
$$t_{m,i} = t_{m,0} + a_1 N_{\max}, \qquad N_i \geq N_{\max} \tag{4.3}$$

where i is the index of the cluster, $t_{m,0}$ is the lower limit for the threshold, a_1 is the increase per segment, N_i is the number of segments in the cluster and N_{max} is the number of segments that is required to achieve the maximum threshold. Again, by taking the same N = 200 female spoken utterances as in previous tests and setting $t_{m,0}$ = 0.5, a_1 = 0.004 and N_{max} = 50, a moving threshold between t_m = 0.5 and t_m = 0.7 was created. As a result of the clustering, the entropy settled to H_c = 0.489, which is worse than with static centroids using t_m = 0.6 (H_c = 0.438). The average number of segments in the clusters had increased from 51 to 54, which is not a significant difference.

By setting the threshold to a higher value $t_{m,0}$ = 0.7 and keeping the other settings as they were, another run was carried out. Now the results were H_c = 0.462, H_s = 0.839 and N_{ave} = 36. In other words, the results were again slightly worse than with a fixed correlation threshold, but the average size of the clusters had decreased. Adaptation was tested once more, now setting t_{m0} = 0.3 and a_1 = 0.008 to gain an extremely large difference in the selectivity between the small and large clusters ($t_{m,min}$ = 0.3, $t_{m,max}$ = 0.9). Entropy grew to H_c = 0.549 and the average cluster size increased to N_{ave} = 88. All results with the adaptive thresholds were worse than with fixed thresholds. This leads to the conclusion that an adaptive threshold does not have a major impact of improving the clustering results.

4.2.3 Clustering experiments using the Multi-level method

The first experiment with Multi-level clustering was meant to determine which order of the frequency band comparisons used as classification criteria gives best results in terms of the selectivity entropy. Identical TIMIT test/female material was used as in the previous method. As described in the previous chapter, data was classified into different sub-spaces at different levels by comparing the correlation of frequency bands from 0 to 1000 Hz, from 1000 Hz to 2000 Hz, and from 2000 Hz to 3000 Hz. A mediocre sized 200-utterance test set was used with fixed thresholds to test all possible combinations. The final level (level 3) entropy was used to measure the quality. Only voiced segments were used in this test. Table 4.12 shows the results. The level of clustering hierarchy where the band is used as classification criteria is shown in the first three columns. A correlation radius threshold t_m = 0.3 was used for all levels.

Table 4.12: Mean entropy of the clusters with different frequency band orders as classification criteria.

0-1 kHz	1-2 kHz	2-3 kHz	H_c	H_s
1	2	3	0.389	0.800
1	3	2	0.389	0.781
2	1	3	0.364	0.802
2	3	1	0.391	0.791
3	1	2	0.356	0.801
3	2	1	0.363	0.782

As the results reveal, it seems that the most efficient order is to first classify by the correlation at the 1000-2000 Hz or the 2000-3000 Hz frequency band, which most often contains the locations of the second and third formant in order. However, the choice between orders [2 1 3] and [3 1 2] was not an obvious one. These combinations were re-evaluated, and now also included the unvoiced segments. Now both methods obtained identical $H_{c213} = H_{c312} = 0.387$ for the phone distributions, while [2 1 3] outperformed [3 1 2] in the cluster size distribution clearly ($H_{s213} = 0.830$ vs. $H_{s312} = 0.782$). Therefore, the frequency band from 1000 Hz to 2000 Hz was chosen to be the decision criteria at the first level, the 0-1000 Hz range at the second level, and the 2000-3000 Hz at the third level.

To obtain a general idea of the clustering process taking place in the hierarchy, voiced-only material ($N = 560$ utterances from female speaker) was clustered. Figure 4.11 shows the cluster entropy levels at the different hierarchy levels when using the same $t_m = 0.3$ correlation threshold for each level. Cluster indices on the x-axis are normalized since the number of clusters increases from level to level.

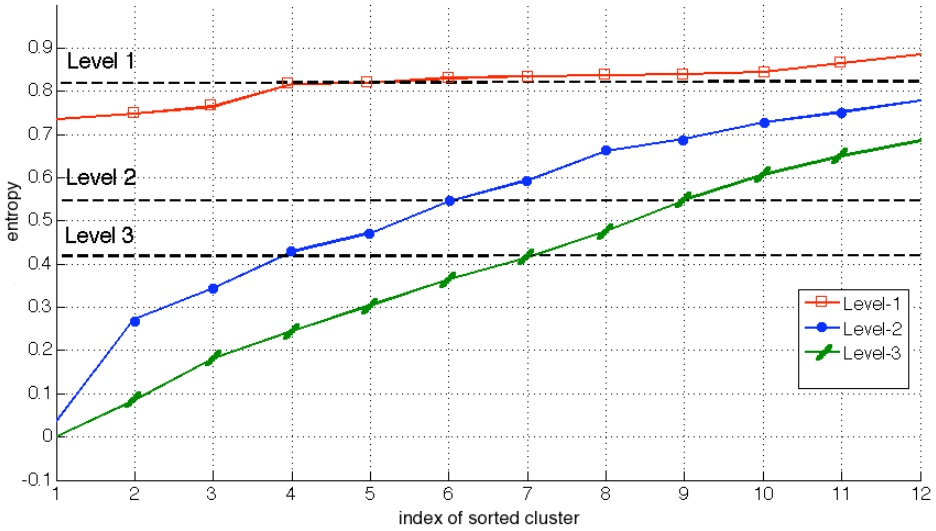

Figure 4.11: Multi-level clustering results, $t_{m1} = t_{m2} = t_{m3} = 0.3$ with level means marked with dashed lines.

As the first two decision criteria contain the usual range of first and second formants, the results support the general understanding in the literature that the first two formants are usually sufficient for differentiating most of the vowels from each other (*Peterson & Barney*, 1952). The final comparison at the 2000-3000 Hz band further decreases the entropy, and manual inspection of the cluster phone distributions seems to indicate that this frequency band seems to help in differentiating many voiced consonants. If the 0-2000 Hz band causes the most noticeable drop in the entropy due to vowel classification, running the test with exactly the same parameters but including also the unvoiced phone segments should lead to a different result regarding the relative entropy changes between levels if they are more dependant on the 2000-3000 Hz frequency-range than voiced phones. Figure 4.12 illustrates what is happening.

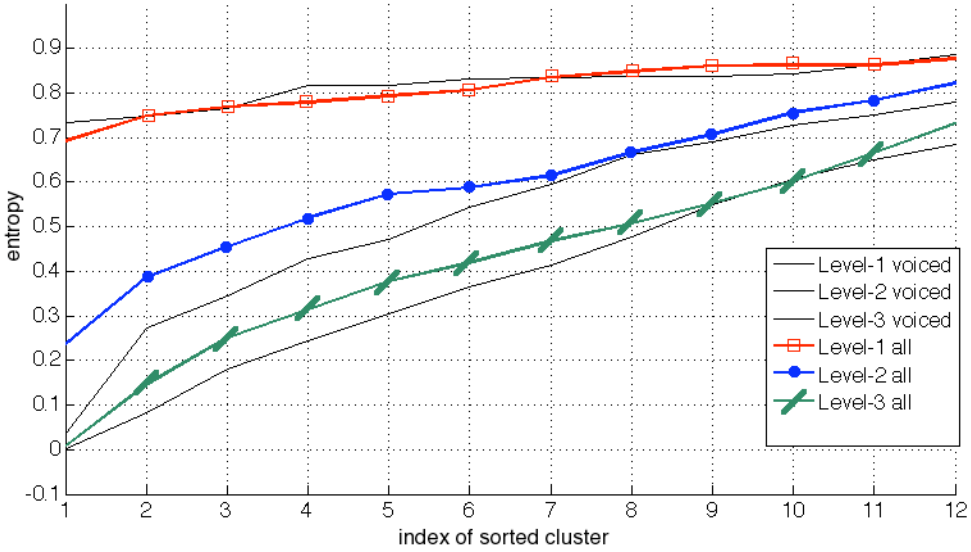

Figure 4.12: Entropies of clusters with both voiced and unvoiced phone segments included. Voiced clustering is shown as grey thin lines for comparison.

The purity of the clusters somewhat degrades, which is expected as the number of segments and variety of the material increases. It should also be recognized that including the unvoiced segments may also include more noisy data, as they are most often classified as unvoiced by cepstrum analysis used for voicing detection. Interestingly, the difference in entropy at the first level is nearly non-existent. Another interesting notion is that the number of very low entropy clusters at level 2 has decreased notably with the unvoiced segments included. Closer inspection reveals that these low-entropy clusters consist mainly of a few fricative segments (e.g., [s]) that have been classified as voiced, one exceptional [d] segment and a handful of nasals. Using otherwise identical conditions, clustering only voiced segments resulted in approximately dH_c = 0.035 better entropy. Since the difference is rather small, the rest of the segmentation test runs were performed with both voiced and unvoiced data together.

The next step was to evaluate the clustering results as a function of the thresholds used at different levels. As the optimization of all possible combinations of three different thresholds is beyond the scope of this work, only the use of identical thresholds with small value adjustments and a few additional extreme cases were tested. Table 4.13 and figure 4.13 show the results for three different thresholds: t_m = 0.1, t_m = 0.3 and t_m = 0.5 for all levels. The t_m parameter shows the correlation threshold for each level and N_{ave} shows the average number of segments per cluster. Table 4.14 shows the results when combinations of different thresholds are used.

As all of the results indicate, the two first steps (F1 and F2) are again very dominating in the classification decisions, splitting the data into several sub-spaces with even lower thresholds. It is difficult to say in which manner the thresholds should be set to gain the best possible classification. Although we can see that increasing t_m will always decrease the entropy (increase cluster purity), the average cluster size will also decrease which

may cause similar phones to be classified into different clusters, and in the worst case into totally different subspaces.

Table 4.13: Results for hierarchical clustering with different thresholds using the same threshold for each level.

Level	t_m	H_c	H_s	N_{ave}
1	0.1	0.864	0.891	3637
2	0.1	0.685	0.838	704
3	0.1	0.591	0.843	156
1	0.3	0.817	0.759	1679
2	0.3	0.604	0.772	248
3	0.3	0.447	0.817	44
1	0.5	0.781	0.739	839
2	0.5	0.485	0.753	83
3	0.5	0.307	0.822	14

Table 4.14: Results for hierarchical clustering with different thresholds for different levels.

Level	t_m	H_c	H_s	N_{ave}
1	0.4	0.805	0.761	1284
2	0.4	0.529	0.749	148
3	0.2	0.404	0.802	37
1	0.2	0.841	0.809	2728
2	0.3	0.633	0.788	352
3	0.4	0.461	0.822	46
1	0.5	0.781	0.739	839
2	0.2	0.608	0.783	182
3	0.5	0.365	0.82	20

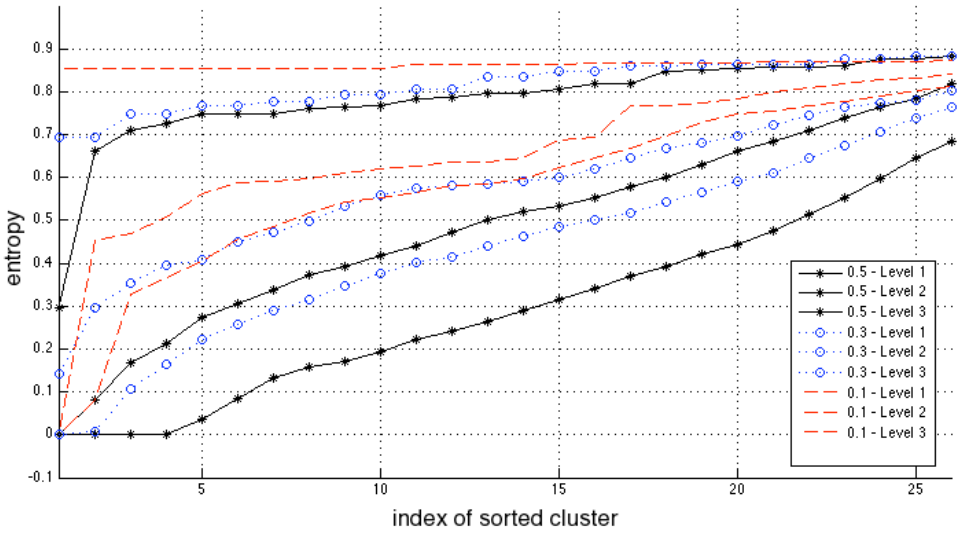

Figure 4.13: Entropies of clusters on different hierarchical levels with different thresholds t_m. The same threshold is used for all three levels ($t_{m,1} = t_{m,2} = t_{m,3}$).

Phone recognition was tested similarly as with the previous clustering method. Recognition of the same material as was used for the original clustering gave a 38 % recognition rate with level 3 clusters for the most dominating labels, and 58 % when using the three most dominating labels. New utterances gave 33 % and 57 % accordingly. These results are very similar to *the Single-Space Method* (38 % with old and 34 % with new speech material), so practically there are no differences in the recognition rates between these two methods.

The differences in the overall classification quality between the *Multi-level Method* and *Single-Space Method* are not large but noticeable (table 4.15). Using t_m = 0.6 with integrative centroids for a single space yields better results than t_m = 0.3 for all levels in hierarchical clustering. While the hierarchical results are slightly less selective, they have a more equal distribution of cluster sizes.

Table 4.15: Comparison of *Single-space* and *Multi-level Methods*.

Type	t_m	N_{tot}	N_{ave}	H_c	H_s
Single-space	0.6	485	45	0.438	0.766
Multi-level	0.3 (all)	496	44	0.448	0.817

One important observation of the *Multi-level Method* is its computational speed: since the segments are always compared to a limited number of clusters in the same sub-space, the number of computational operations needed to find the best matching cluster is much less than in the *Single-space Method* when data volumes grow large. Memory requirements are also reduced, since only the relevant sub-space information needs to be kept in memory during operations and all other sub-spaces can be discarded. For example, the time needed for a 0.8 correlation threshold clustering of 560 utterances with the *Single-space Method* in the MATLAB environment with a relatively powerful Mac Pro takes approximately 20 minutes, while the same mean entropy clustering can be completed in 5 minutes with the *Multi-level Method*. The difference becomes larger if the amount of data is increased, although the computational complexity in the both methods should converge to a specific level when all possible locations in the cluster space are covered. However, using lower thresholds (t_m< 0.7) and cleaning the cluster space frequently to keep the number of clusters limited leads to acceptable computational times with the *Single-Space Method* over large sets of material as well. Despite the more efficient structure of the hierarchical method, there is also a maximum value for the merge criterion, after which the cluster space structure starts to expand with a remarkable rate, turning the classification process into a dispersion process.

4.2.4 Feature extraction and parameter evaluation

The effects of different parameters in feature extraction were tested in order to justify preliminary parameter selections. Central parameters in the feature extraction algorithm are the number n_s of spectral vectors used to form the feature vector, the size of the margins m_1 and m_2 at both ends of the segment, and the location of the divisor d for dividing the segment into onset and offset parts (fig. 4.14). It was originally hypothesized

in the previous chapter that placing the parameter d at the 40 % duration point of the segment would lead to a relatively pure phone description of the first part of the segment, and to a more context-dependant second part. To confirm this, the location of d as a proportion of the total duration of the segment was adjusted in the feature extraction process and then a single-space clustering was performed to measure changes in the entropy H_c. 200 female spoken TIMIT signals (a total of 7700 segments) were used for these experiments. Margins m_1 and m_2 were kept at their default value m = 10 %. Table 4.16 shows the results.

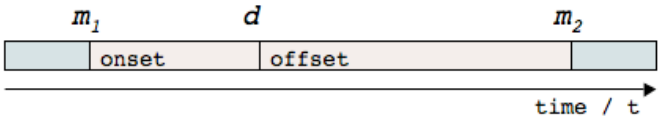

Figure 4.14: Division of the segment into subsegments.

Table 4.16: Effect of the segment divisor location on cluster selectivity.

d (%)	15	20	30	40	60	80
H_c	0.460	0.470	0.500	0.496	0.492	0.49

According to these results it seems that the first 15-20 % of the segment contains the best static spectral description of the segment. Going below d = 15 % with 10 % margins is not practical since the number of pitch-synchronous windows starts to be insufficient compared to the number of spectral vectors taken for feature representation. The minimum entropy is not located near the vicinity of the 40 % boundary, which was the preliminary assignment for the parameter d due to empiric observations. To gain a more comprehensible understanding of the segmental division, the effects of margin size on the entropy were also evaluated (table 4.17).

Table 4.17: Effect of margin size in feature extraction to cluster selectivity.

m (%)	1	3	5	7	10
H_c	0.517	0.49	0.468	0.483	0.496

The symbol m describes the portion of the segment duration that is ignored from the beginning and from the end of the segment in feature extraction to avoid noisy or non-contrasting spectral vectors. It seems that the feature vectors are most distinct when the margin is approximately 5 % of the segment duration. Using very short or even non-existent margins causes the entropy to increase as expected, since some of the spectral vectors may be taken from the transition points between two subsequent phones instead of the locus of a single phone. Increasing the margin size above 5% also hinders the classification accuracy, an observation being in line with the results above concerning segment division.

The last central issue in feature extraction is the number of spectral vectors that are taken and averaged to form one single spectral representation for each segment. The same experimental arrangements were used to estimate the selectivity of the clusters as above. Margins were set to 5 % and segments were divided at d = 40 % to gain a capture region of 35 % of the segment duration for the spectral vectors.

Table 4.18: Effect of the number of spectral vectors n_s, female speakers.

n_s	1	2	3	4	5	7	12
H_c	0.477	0.459	0.491	0.483	0.460	0.483	0.510

The value $n_s = 5$, that was chosen in advance as an enlightened guess and was used in all previous experiments, indicates that it is at the same level in entropy as other low-entropy selections for n_s (table 4.18). Interpretation of these results is not straightforward, though. It seems that $n_s = 2$ and $n_s = 5$ produce almost the same selectivity for the clusters, while $n_s = 3$ and $n_s = 4$ yield a relatively high entropy compared to previous ones. To determine whether the phenomenon noticed with female speakers recurs with male spoken material, $N = 200$ male utterances from the TIMIT corpus were tested in a similar manner.

Table 4.19: Effect of the number of spectral vectors n_s, male speakers.

n_s	1	3	5	7	10
H_c	0.500	0.507	0.519	0.510	0.515

Together these results (tables 4.18 & 4.19) seem to indicate that using a low number of spectral vectors may provide a small advantage over a larger number of vectors ($n_s > 5$), but as the dependency is not directly observable and rather ambiguous, nothing definite can be concluded.

4.2.6 Conclusions on clustering and feature extraction experiments

Two structurally different approaches to segmental data classification were tested. In terms of the entropy and average cluster size, the results are very similar for both methods. However, the manner in which the algorithms obtain the final state have a fundamental difference: the *Multi-level method* achieves the final state directly with a strict division to three levels of subspaces, each refining the clustering quality by comparing separate frequency bands and without performing further merging or cleaning operations to small or closely located clusters. On the contrary, the *Single-Space method* grows a very diverse amount of small clusters, which are later merged into larger ones if they seem to be one of a kind with large amounts of speech material. This may be a problem with learning new material, where there exist allophones that occur very rarely but should be nonetheless conserved as autonomic classifications. However, recognition tests did not show changes in the results when small clusters were merged away. It is also noteworthy that the classification of all material in the hierarchical experiments were performed only by scalar product based distance metrics and with a 0-3000 Hz frequency band comparison that is most suitable for the classification of vowels. The structure of this method essentially supports any quantitative decision criteria at any level, and therefore a search for more reliable decision criteria can be profitable in the future.

The feature extraction experiments indicate that in order to describe phone segments with spectral coefficients averaged from the segment, the best place to pick these vectors is from the very beginning of the phone segment. It is recommended to use margins of

approximately 5 % of the segment duration in the beginning of the segment to prevent selection of spectral vectors from the transition points.

Most results from the clustering experiments do not directly describe how well the methodology suits purposes of speech recognition by learning. The major goal of these experiments was to compare different experimental approaches to segmental data classification problem and to justify and re-evaluate preliminary chosen solutions and parameters. The recognition rates for phones were also tested, reaching approximately 30 % for the top label and 60 % for the three most dominating labels for each cluster with a relatively rough clustering. Naturally, this is nowhere near trained HMM based phone recognizers that usually obtain correct recognition rates of approximately 75 % of phones (see *Petrov et al.*, 2007). However, it should be kept in mind that the purpose of this project was not to build a phone recognizer, nor should it be evaluated solely on the basis of phone recognition. The ultimate effect of the chosen solutions for classification will be evaluated in terms of its compatibility and performance with a larger unsupervised learning recognition system that it will be integrated into.

One issue worth discussing is that most of the evaluations used in this thesis rely very strongly on the entropy value of the phone distribution and the average size of the clusters. The entropy shows the selectivity of an average distribution of phones in each cluster, but it does not rely on a phonetic basis for quality, i.e., it does not directly depict how well the clusters actually classify different aspects of speech units that can be used for speech recognition purposes. It is also highly dependent on the segmentation algorithm that provides the data for classification, and the reference that is used for labeling. Therefore, other supportive methods for clustering evaluation could turn out to be useful in terms of future development of the methodology.

4.3 Word recognition

An experiment to test the functionality of the system developed so far was a certain type of word recognition, or, pattern discovery test. Utterances containing the same key words were used as an input to the system, and then cluster indices of the segments (or *labels*) were compared manually to see whether similar words follow similar paths in the cluster space. The idea was not to build sufficient statistics for a comprehensive evaluation of the patterns formulated by the algorithms, but to obtain a quick review of what has been achieved so far.

Word recognition experiments were done with the TIDIGITS corpus (*Leonard & Doddington*, 1993), which contains English spoken utterances consisting of connected digit sequences. Signals were resampled from 20 kHz to 16 kHz in order to match the input specifications of the system. No phonetic transcription was available to be used in a phonetic label comparison or segmentation quality evaluation. Tests were carried out

with gender specific material, i.e., in this case only the female set was used[12]. However, the material still contained several different speakers. The training set (N = 4388 utterances) was used for forming the cluster space and the test set was used for testing the labeling. Single-space clustering with $t_m = 0.5$ was used with merging of all of the clusters closer than $t_{merg} = 0.5$ to each other as a post-processing step before moving on to actual test utterances. This pre-clustering resulted in $N_{tot} = 22$ separate clusters.

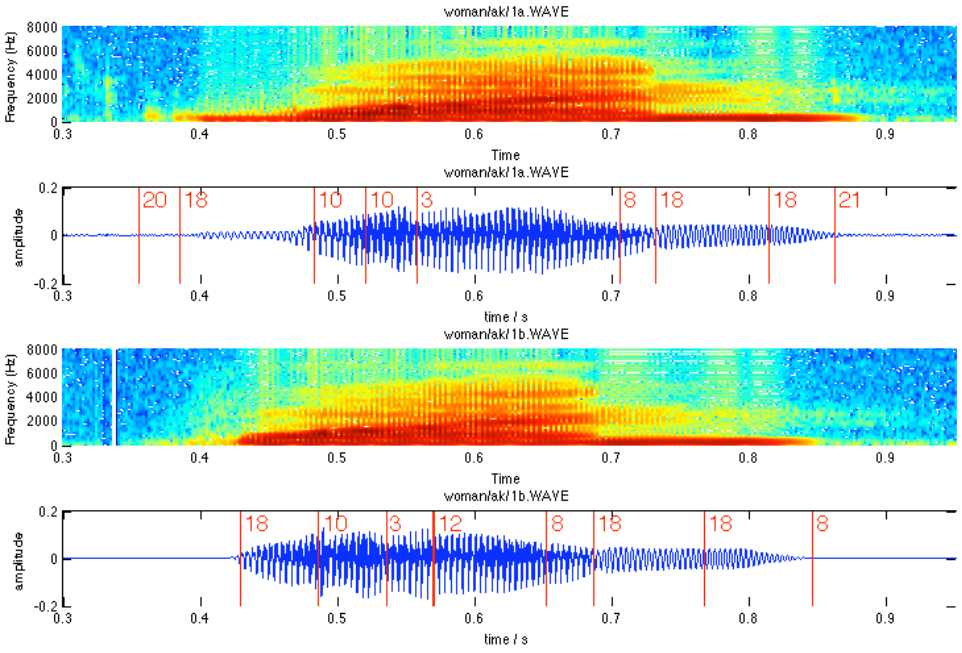

Figure 4.15: Clustering labels of two different realizations *1a* and *1b* of the word *"one"* spoken by a female speaker.

[12] TIDIGITS is divided into a *train* and *test* set similarly to TIMIT, both containing 4 subsets of utterances: *boy*, *girl*, *man*, and *woman*.

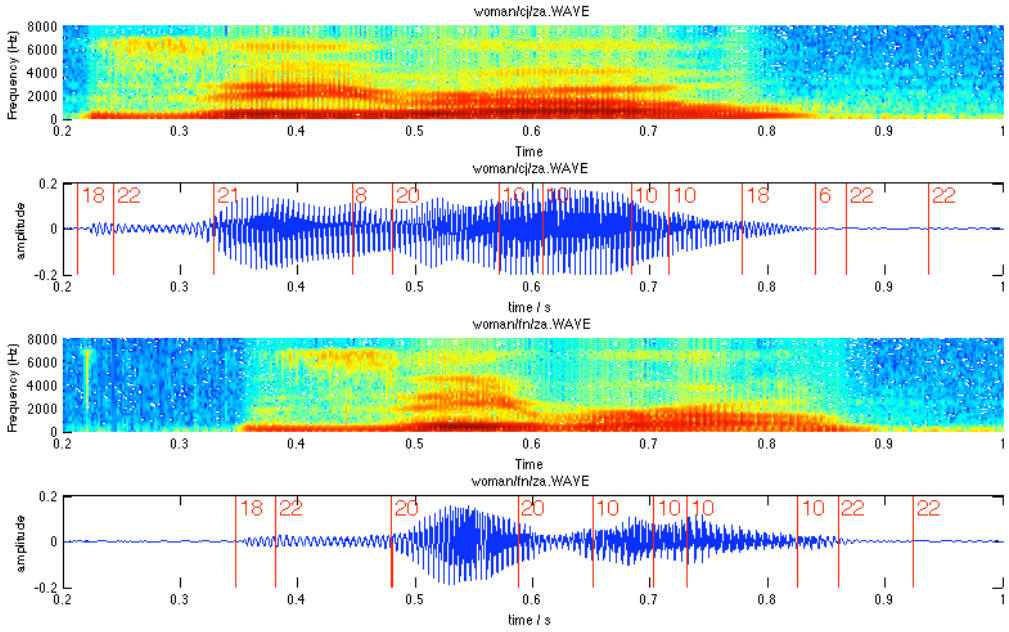

Figure 4.16: Clustering labels of two different realizations of the word *"zero"* spoken by two different female speakers.

The first test was performed with two different realizations of a single digit word, *"one"*, spoken by the same female speaker. As can be seen from the image (fig. 4.15, segment boundaries and cluster indices are marked with red), several segments of the word are classified similarly. The only difference (excluding the silence) is the splitting of the middle section of the word into two different parts in signal *1b*, causing it to be clustered into clusters $i = 3$ and $i = 12$, instead of just $i = 3$ as in utterance *1a*.

To increase the level of difficulty, the words *"zero"* were compared from two completely different sounding female speakers. The major structure (fig. 4.16) of the clustering was still similar, while there were also some differences. As can be seen from the signal waveform and the spectrogram, these two realizations have very different temporal structure. There are also subtle differences in the frequency representations, concerning, e.g., absolute formant frequencies and relative formant energies.

The next experiment was to compare two utterances, *"one-five-seven"* and *"seven-one-one"* to get a preliminary grasp of how word location in the utterance affects classification. Figure 4.17 shows the spectrograms and waveforms for these two utterances.

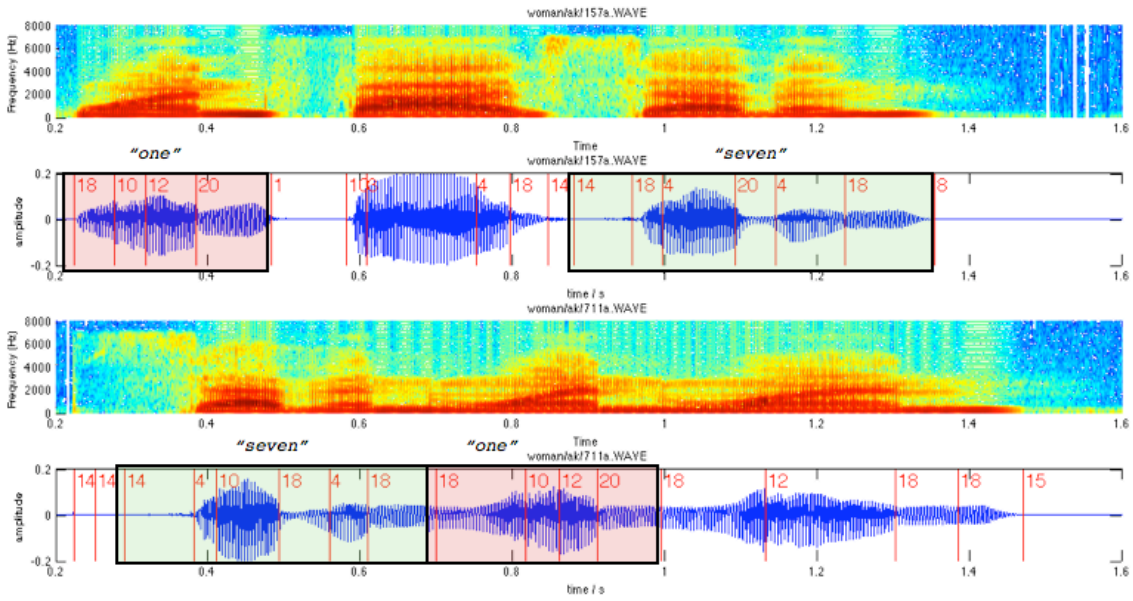

Figure 4.17: Utterances *"one-five-seven"* (*157a*) and *"seven-one-one"* (*711a*). The most similar pair of *"one" words* is marked with red squares while *"seven"* words are marked with green squares.

Two *ones* are segmented with identical structure and with identical labels, but the third realization at the end of the utterance *711a* differs somewhat from these two. The *sevens* also have common clusters in the cluster sequence but several differences exist. When interpreting these results, it should be kept in mind that the clustering precision is very rough (t_m = 0.5), meaning that the classification accuracy is also rather limited. On the other hand, the clustering is done with several speakers without any speaker normalization, which is a more difficult situation than with single speaker experiments.

To remove the effects of several speakers and the requirement to use pre-existing clusters, a final test was arranged. Figure 4.18 shows the results for the same utterances as above when no pre-existing clustering with a training set was used. This means that the cluster space is in this case built incrementally, segment by segment, using only these two signals. Now the similarity has increased especially in the case of *sevens*. Also the *ones* have 92 % (11/12) congruence in the structure and labeling, if boundary insertion due to spectral fading is ignored in the last realization of *711a*. The small difference is caused by a segment boundary deletion at approximately t = 1.1 s in the same utterance.

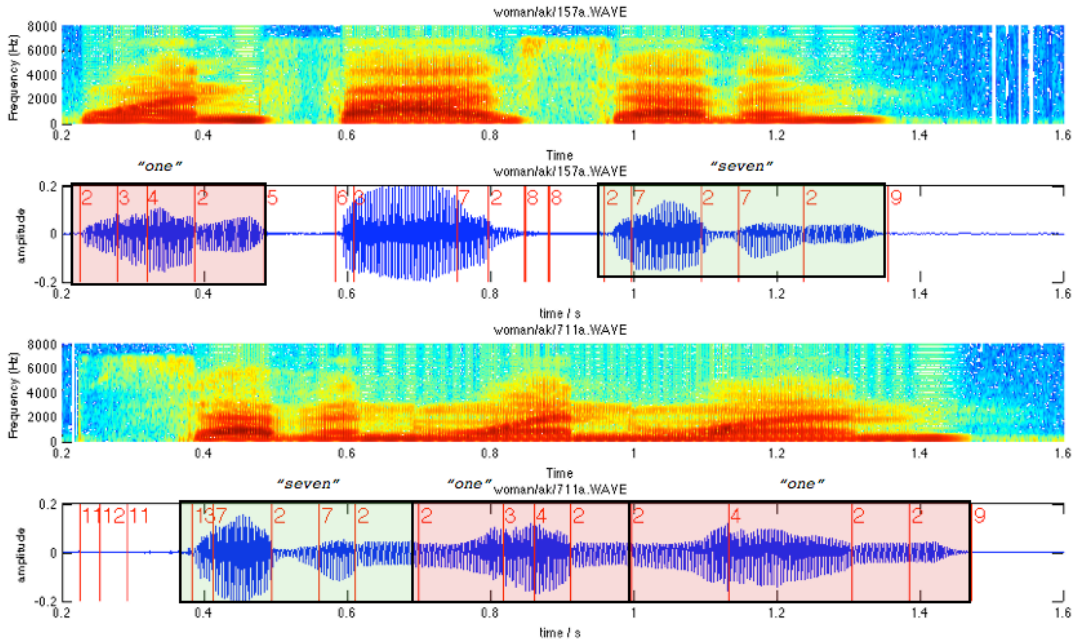

Figure 4.18: Utterances *"one-five-seven"* (*157a*) and *"seven-one-one"* (*711a*) without pre-clustering.

While these small experiments of "word recognition" do not give a statistically sufficient understanding of the functionality of the system, the results are a glimpse of the current status considering future work. The classification is not perfect, nor is it supposed to be at this very initial phase of the entire project of building a learning system, where the learning structures themselves are still only sketches on a blackboard. Nonetheless, the results are positive and show that the system is already capable of converting simple acoustic speech signals into structured representations containing only temporal boundaries and labels for each corresponding segment.

5 Conclusions and future directions

The long-term goal of the ACORNS research project, in which this work contributes to, is to develop a learning speech recognizer. The recognizer will be able to build statistical representations of speech by bottom-up processing the input and then refining the bottom-up process itself by intelligent top-down feedback. In order to function properly, this speech recognition architecture requires a strong methodological bottom-up basis to provide consistent descriptions of the speech signals. To support such a need, novel methodological approaches were designed, implemented and evaluated within this thesis. Incoming speech is first segmented into phone-sized units with a blind segmentation algorithm. Then feature-based descriptions of each segment are extracted with a feature extraction algorithm, and finally, the segmental data is classified by using specific clustering techniques.

The segmentation algorithm introduced in this thesis utilizes cross-correlation estimates of short window length FFT-spectrums in order to detect potential phonetic boundaries. Two-dimensional filtering of the cross-correlation matrix and a subsequent non-linear filtering process with a peak masking operation enables refined and robust segment boundary detection. The algorithm achieved good results in comparison to results reported in current literature. It was also demonstrated that the segmentation process could be performed in extremely noisy conditions, with the observation that locations of actual segment boundaries may often be found by unintentional random hypotheses instead of actual spectral change detections. Concerning this observation, different aspects of the segmentation evaluation were also discussed in more detail.

Concerning data classification, the clustering methodology used in the architecture has a few important requirements: the nature of learning is incremental, and so is the availability of speech data. Therefore, the clustering algorithm must be able to work in an incremental manner instead of just performing batch processing. It is also an important aspect of the learning system that the maximum amount of data that can be potentially classified or learned should not be limited. This sets specific requirements for the design and implementation of the clustering algorithm. Two different clustering algorithms based on different structures for the cluster spaces were introduced and evaluated. Solutions and parameter selections used in feature extraction were also evaluated in order to justify, or, to disqualify the preliminary assumptions. Shannon entropy of the cluster selectivity, the entropy of the distribution of the cluster sizes, and phone recognition rates

were used as a central criteria in the evaluation. However, the ultimate quality of these methodological solutions will only be seen in terms of the functionality of the entire speech recognition system.

The major goal of future research is to search for possible mechanisms that can be employed to enable top-down feedback and ultimately pattern discovery by learning. Methodological solutions for representing segmented utterances at an abstract level, and gathering statistics on speech signals as trajectories in the cluster space, may enlighten the path. It may be also intriguing to find out how the existing computational models of human memory (e.g., MINERVA2) perform on segmental statistics instead of direct acoustic features.

It may also be useful to study the characteristics of the current segmentation algorithm in more detail. By building labeled statistics of the phonetic boundaries between the segments, and by comparing the transitions across different pairs of phones, more insight should be available to understand the problems and strengths of the algorithm. Careful manual inspection of the problematic points in the speech stream and well-covered tools to gain a deeper understanding about the behavior of the algorithm at those points will also provide valuable information. Understanding the structural and phenomenal reasons behind the flaws in the segmentation and clustering processes may indicate what sort of feedback is required from the higher levels. This also leads to a central question of the entire speech recognition theme: what is the composition of predictions and feedbacks that need to be made at the different levels of processing in order to optimize the learning process and recognition performance?

Advances in the incremental clustering methodology will also be a topic of great interest. Incremental construction of the cluster space, utterance-by-utterance from an initially empty set, is not a deterministic process in a manner such that the same configuration of clusters, despite the order in which data is fed, would be attained. For example, it may be interesting to observe what happens if copies of the incoming segments are stored into separate storage, while incremental clustering is performed in parallel. Then later on, the contents of this storage could be used as new input to the clustering system that could be run in a batch mode in order to modify or consolidate the organization of the incrementally built space. This may lead to a more optimized organization of the clusters and enable learning of new classification structures (clusters), instead of only trying to optimize the input to existing structures. By using a little imagination, one could compare this to human cognitive processing during the waking hours and the re-organization and consolidation of memory taking place during sleep.

References

Ajmera J., McCowan I. & Bourlard H.: *Robust Speaker Change Detection.* IEEE Signal Processing Letters, Vol. 11, No. 8, 2004

Antal M.: *Speaker Independent Phoneme Classification in Continuous Speech.* Studia Univ. Babes-Bolyal, Informatica, Vol. 49, No. 2, 2004

Aversano G., Esposito A., Esposito A. & Marinaro M.: *A New Text-Independent Method for Phoneme Segmentation*. Proceedings of the IEEE international Workshop on Circuits and Systems, Vol. 2, pp. 516-519, 2001.

Balducci F. & Cerrato L.: *Performance Evaluation Test of an Automatic Segmentation System for Italian and American-English Continuous Speech.* Proceedings of the International Conference of Phonetic Sciences, S. Francisco, 1999

Barry W.J.: *SALA labeling tests.* Summary report, SAM-document: SAM UCL-020, 1991

Benedict H.: *Early lexical development: Comprehension and production.* Journal of Child Language, Vol. 6, pp. 183-200, 1979

Bishop C.M.: *Pattern Recognition and Machine Learning.* Springer, 2007

Blumstein S.E., Mehler J., Bertoncini J. & Bilejljac-Babic R.: *Discrimination in neonates of very short CVs.* Journal of Acoustical Society of America, Vol. 82, 1987

Boatman, D.: *Cortical bases of speech perception: evidence from functional lesion studies.* Cognition, Vol. 92, pp. 47-65, 2004

Boves L., ten Bosch L. & Moore R.: *ACORNS – Towards computational modeling of communication and recognition skills.* Proceedings of the IEEE international Conference on Cognitive Informatics (ICCI'07), 2007.

Buonomano, D.V. & Merzenich M.M.: *Cortical Plasticity: From Synapses to Maps.* Annual Review of Neuroscience, Vol. 21, pp. 149-86, 1998

Burton M.W., Small S.L. & Blumstein S.E.: *The role of Segmentation in Phonological Processing: An fMRI Investigation.* Journal of Cognitive Neuroscience, Vol. 12, pp. 679-690, 2000

Collins A.M. & Loftus E.F.: *A spreading-activation theory of semantic processing.* Psychological Review, Vol. 82, pp. 407-428, 1975

Collins A.M. & Quillian M.R.: *Retrieval time from semantic memory.* Journal of Verbal Learning and Verbal Behavior, Vol. 8, pp. 240-247, 1969

Cucchiarini C. & Strik H.: *Automatic Phonetic Transcription: An overview.* Proceedings of 15th International Congress of Phonetic Science (ICPhS), Barcelona, Spain, pp. 347-350, 2003

Deller J.R., Hansen J.H.L. & Proakis J.G.: *Discrete-Time Processing of Speech Signals.* IEEE Press, 2000

Demuynck K. & Laureys T.: *A Comparison of Different Approaches to Automatic Speech Segmentation.* Proceedings of the 5th International Conference on Text, Speech and Dialogue (TSD), pp. 277-284, 2002.

Duda R.O., Hart P.E. & Stork D.G.: *Pattern Classification (2nd ed.).* John Wiley and sons, 2001

Elliot L.L., Busse L.A., Partridge R., Rupert J. & DeGraaff R.: *Adult and Child Discrimination of CV Syllables Differing in Voicing Onset Time*. Child Development, Vol. 57, No. 3, pp. 628-635, 1986

Erdmann C., Vary P., Fischer K., Stegmann J., Quinquis C., Massaloux D. & Kovesi B.: *An adaptive multi rate wideband speech codec with adaptive gain re-quantization*. Proceedings of IEEE Workshop of Speech Coding, pp. 145-147, 2000

Estevan Y.P., Wan V. & Scharenborg O.: *Finding Maximum Margin Segments in Speech.* Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing 2007 (ICASSP '07), Vol. 4, pp. 937-940, 2007

Fitch W.T.: *The evolution of speech: a comparative review*. Trends in cognitive sciences, Vol. 4, No. 7, pp. 258-267, 2000

Garofolo J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G. & Pallett, D.S.: *DARPA TIMIT acoustic-phonetic continous speech corpus*, Unknown, 1993

Gazzaniga M., Ivry R.B. & Mangun G.R.: *Cognitive Neuroscience: the Biology of the Mind, 2nd ed.* W. W. Norton & Company, 2002

Greenberg S.: *Strategies for automatic multi-tier annotation of spoken language corpora.* In proceedings of the 8th European Conf. on Speech Communication and Technology (Eurospeech-2003), pp. 45–48, 2003

Gauker C.: *No conceptual thinking without language.* Behavioral & Brain Sciences, Vol. 25, Issue 6, pp. 687, Dec. 2002

Hagen A., Connors D.A. & Pellm B.L.: *The Analysis and Design of Architecture Systems for Speech Recognition on Modern Handheld-Computing Devices.* Proceedings of the 1st IEEE/ACM/IFIP international conference on hardware/software design and system synthesis, pp. 65-70, 2003

Handl J. & Knowles J.: *An Evolutionary Approach to Multiobjective Clustering.* IEEE Transactions on Evolutionary Computation, Vol. 11, No. 1, 2007

Hawkins, J. & Blakeslee S.: *On Intelligence. How a new understanding of the brain will lead to the creation of truly intelligent machines.* Published by arrangement with Sane Töregård Agency AB, 2004

Hermansky, H.: *Perceptual linear predictive (PLP) analysis for speech.* Journal of Acoustic Society of America, Vol. 87, pp. 1738-1752, 1990

Hermansky, H.: *Should Recognizers Have Ears?* Speech Communication, Vol. 1998, No. 25, pp. 3-27, 1998

Hermansky H., Morgan N., Bayya A. & Kohn P.: *RASTA-PLP Speech Analysis.* Technical Report (TR-91-069), International Computer Science Institute, Berkeley, CA., 1991

Hintzman D.L.: *MINERVA 2: A simulation model of human memory.* Behavior Research Methods, Instruments & Computers, Vol. 16, pp. 96-101, 1984

Hintzman D.L.: *"Schema abstraction" in a multiple-trace memory model.* Psychological Review, Vol. 93, pp. 411-428, 1986

Ishizuka K. & Nakatani T.: *A feature extraction method using subband based periodicity and aperiodicity decomposition with noise robust frontend processing for automatic speech recognition.* Speech Communication, Vol. 48, Issue 11, pp. 1447-1457, 2006

Jain A.K., Murty M.N. & Flynn P.J.: *Data Clustering: A Review.* ACM Computing Surveys, Vol. 31, pp. 264-323, 1999

Juang B.H. & Rabiner L.R: *Automatic Speech Recognition – A Brief History of The Technology Development.* Elsevier Encyclopedia of Language and Linguistics, Second Edition, 2005

Karjalainen M.: *Kommunikaatioakustiikka*. Helsinki University of Technology Laboratory of Acoustics and Audio Signal Processing, Technical Report 51, 1999

Kemp, D.T.: *Stimulated acoustic emissions from within the human auditory system.* Journal of Acoustical Society of America, Vol. 64, pp. 1386-1391, 1978

Kim J., Kim S., Kim D., Lee W. & Kim E.: *Low-Energy Localized Clustering: An Adaptive Cluster Radius Configuration Scheme for Topology Control in Wireless Sensor Networks.* Vehicular Technology Conference, 2005. IEEE 61st. Vol. 3, pp. 2546-2550, 2005

Knill K. & Young S.: *Hidden Markov Models in Speech and Language Processing.* In Young S. and Bloothooft G. (eds.), Corpus-Based Methods in Language and Speech Processing. Kluwer Academic Publishers, pp. 27-68, 1997

Kosaki H., Hashikawa T., He J. & Jones E.G.: *Tonotopic organization of auditory cortical fields delineated by parvalbumin immunoreactivity in macaque monkeys.* Journal of Comparative Neurology, Vol. 386, pp. 304-316, 1997

Kvale K.: *Segmentation and Labeling of Speech.* PhD Dissertation, The Norwegian Institute of Technology, 1993

Ladefoged P.A.: *A Course in Phonetics.* Harcourt Brace Jovanovich, Publishers, 1982

Laver J.: *Principles of Phonetics.* Cambridge, Cambridge University Press, 1994

Lee Y. & Hwang K.-W.: *Selecting Good speech Features for Recognition.* ETRI Journal, Vol. 18, No. 1, 1996

Leonard R.G. & Doddington G.: *TIDIGITS.* Linguistic Data Consortium, Philadelphia, 1993

Li T.H. & Gibson J.D.: *Speech Analysis and Segmentation by Parametric Filtering.* IEEE Transactions on speech and audio processing, Vol. 4, No. 3, pp. 203-13, May 1996

Lieberman A.M. & Mattingly I.G.: *The motor theory of speech perception revisited.* Cognition, Vol. 21, pp. 1-36, 1985

LoCasto P.C., Krebs-Noble D., Gullapalli R.P. & Burton M.W.: *An fMRI Investigation of Speech and Tone Segmentation.* Journal of Cognitive Neuroscience, Vol. 16, pp. 1612-1624, 2004

Maier V. & Moore R.K.: *An investigation into a Simulation of Episodic Memory for Automatic Speech Recognition.* Proceedings of ISCA INTERSPEECH, pp. 1245-1248, 2005

Mcgurk H. & MacDonald J: *Hearing lips and seeing voices.* Nature, Vol. 264, pp. 746-748, 1976

McLelland J.L. & Elman J.L.: *The TRACE Model of Speech Perception.* Cognitive Psychology, Vol. 18, pp. 1-86, 1986

McNealy K., Mazziotta J.C. & Dapretto M.: *Cracking the Language Code: Neural Mechanisms Underlying Speech Parsing.* The Journal of Neuroscience, Vol. 26, pp. 7629-7639, 2006

McQueen J.: *Some methods for classification and analysis of multivariate observations.* Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967

Mermelstein P.: *Automatic segmentation of speech into syllabic units*. Journal of Acoustical Society of America, Vol. 58, No. 4, pp. 880-883, Oct. 1975.

Moore B.C.J.: *Hearing.* Academic Press, San Diego, 1995

Moore R.K. & Maier V.: *Preserving fine phonetic detail using episodic memory: Automatic speech recognition with MINERVA2.* Proceedings of 16th International Congress of Phonetic Science (ICPhS)**,** 2007

Motlíček P.: *Feature Extraction in Speech Coding and Recognition*, Report, Portland, US, Oregon Graduate Institute of Science and Technology, pp. 1-50, 2002

Morais J., Cary L., Alegría J. & Bertelson P.: *Does awareness of speech as a sequence of phones arise spontaneously?* Cognition, Vol. 7, pp. 323-331, 1979

Morais J., Bertelson P., Cary L. & Alegría J.: *Literacy training and speech segmentation.* Cognition, Vol. 24, pp. 45-64, 1986

Mountcastle V.B: *An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System,* in *The Mindful Brain (*ed. Edelman G.M. & Mountcastle V.B.). MIT Press, Cambridge, Massachusetts, 1978

Neath I. & Surprenant, A.M: *Human memory: an introduction to research, data, and theory.* Belmont, CA: Thomson/Wadsworth, 2003

Nishitani N. & Hari R.: *Viewing Lip Forms: Cortical Dynamics.* Neuron, Vol. 36, pp. 1211-1220, 2002

Norris D., McQueen J. M. & Cutler A.: *Perceptual learning in speech.* Cognitive Psychology, Vol. 47, pp. 204-238, 2003

Parikh G. & Loizou P.C.: *The influence of noise on vowel and consonant cues.* Journal of Acoustical Society of America, Vol. 118, No. 6, pp. 3874-3888, 2005

Park A. & Glass J.R.: *Unsupervised word acquisition from speech using pattern discovery.* Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing 2006 (ICASSP '06), Vol. 1, 2006

Petek B., Andersen O. & Dalsgaard P.: *On the Robust Automatic Segmentation of Spontaneous Speech.* In Proceedings of ICSLP '96, pp. 913-916, 1996

Peterson G.E. & Barney H.L.: *Control Methods Used in a Study of the Vowels.* The Journal of the Acoustical Society of America, Vol. 24, No. 2, 1952

Petrov S., Pauls A. & Klein D.: *Learning Structured Models for Phone Recognition.* Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 897-905, 2007

Price C., Thierry G. & Griffiths T.: *Speech-specific auditory processing: where is it?* Trends in Cognitive Sciences, Vol. 9, No. 6, 2005

Rabiner L.R. & Schafer R.W.: *Digital Processing of Speech Signals,* Prentice-Hall, 1978

Rizzolatti G. & Craighero L.: *The mirror-neuron system.* Annual Review of Neuroscience, Vol. 27, pp. 169-192, 2004

Roy D.: *Grounding words in perception and action: computational insights.* Trends in Cognitive Sciences, Vol. 9, No. 8., pp. 389-396, 2005

Saffran J.R., Aslin R.N. & Newport E.L.: *Statistical Learning by 8-Month-Old Infants.* Science, Vol. 274, pp. 1926-1928, 1996

SaiJayram A.K.V, Ramasubramanian V. & Sreenivas T.V.: *Robust parameters for automatic segmentation of speech*. Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing 2002 (ICASSP '02), Vol. 1, pp. 513-516, 2002

Samuel A.G.: *Phonemic restoration: Insights from a new methodology.* Journal of Experimental Psychology: General, Vol. 110, pp. 474-494, 1981

Sarkar A. & Sreenivas T.V.: *Automatic speech segmentation using average level crossing rate information*. Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing 2005 (ICASSP '05), Vol. 1, pp. 397-400, 2005

Scharenborg O., Ernestus M. & Wan V.: *Segmentation of speech: Child's play?* Proceedings of Interspeech, 2007

Schiel F.: *Automatic Phonetic Transcription of Non-Prompted Speech.* Proceedings of the ICPhS 1999. San Francisco, August 1999. pp. 607-610, 1999

Sharma M. & Mammone R.: *"Blind" speech segmentation: Automatic segmentation of speech without linguistic knowledge.* Spoken Language, 1996. ICSLP 96. Proceedings. Vol. 2, pp. 1237-1240, 1996

Smith E.E., Shoben E.J. & Rips L.J: *Structure and process in semantic memory: A featural model for semantic decisions.* Psychological Review, Vol. 81, pp. 214-241, 1974

Stager C.L. & Werker J.F: *Infants listen for more phonetic detail in speech perception than in word-learning tasks.* Nature, Vol. 388, 1997

Stefanatos G.A., Joe W.Q., Aguirre G.K., Detre J.A. & Wetmore G.: *Activation of human auditory cortex during speech perception: Effects on monaural, binaural and dichotic presentation.* Neuropsychologia, 2007

Swingley D.: *Statistical clustering and the contents of the infant vocabulary.* Cognitive Psychology, Vol. 50, pp. 86-132, 2005

Toledano, D.T., Hernández Gómez L.A. & Grande L.V.: *Automatic Phonetic Segmentation.* IEEE Transactions on Speech and Audio Processing, Vol. 11, No. 6, 2003

Toro J.M., Sinnet S., Soto-Faraco S.: *Speech segmentation by statistical learning depends on attention.* Cognition, Vol. 97, pp. B25-B34, 2005

Trehub S.E.: *The Discrimination of Foreign Speech Contrasts by Infants and Adults.* Child Development, Vol. 47, No. 2, 1976

Ulfendahl M. & Flock Å.: *Outer Hair Cells Provide Active Tuning in the Organ of Corti.* News in Physiological Sciences, Vol. 13, 1998

Wade T., Eakin D.K., Webb R., Agah A., Brown F., Jongman A., Gauch J., Schreiber T.A. & Sereno J.: *Modeling recognition of speech sounds with MINERVA2.* Proceedings of ICSLP, pp. 1653-1656, 2002

Warren R.M.: *Perceptual restoration of missing speech sounds.* Science, Vol. 167, pp. 392-393, 1970

Werker J.F. & Tees R.C.: *Cross-language speech perception: Evidence for perceptual reorganization during the first year of life.* Infant Behavioral Development, Vol. 7, pp. 49-63, 1984

Wesenick M.-B. & Kipp A.: *Estimating the Quality of Phonetic Transcriptions and Segmentations of Speech Signals.* Proceedings of ICSLP, pp. 129-132, 1996

Wilson S.M., Saygin A.P., Sereno M.I. & Iacoboni M.: *Listening to speech activates motor areas involved in speech production.* Nature Neuroscience, Vol. 7, No. 7, 2004

Wissinger C.M., VanMeter J., Tian B., Van Lare J., Pekar J. & Raushecker J.P.: *Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging.* Journal of Cognitive Neuroscience, Vol. 13, pp. 1-7, 2001

Zahorian S.A., Silsbee P. & Wang X.: *Phone classification with segmental features and a binary-pair partitioned neural network classifier.* Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97), Vol. 2, pp. 1011-1014, 1997

Zhang T. & Kuo C.-C. J.: *Hierarchical classification of audio data for archiving and retrieving.* Proceedings of the Acoustics, Speech, and Signal Processing 1999 on 1999 IEEE International Conference, Vol. 6, pp. 3001-3004, 1999

Appendix A

2nd order pre-emphasis filter

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

$b_0 = 0.3426$
$b_1 = 0.4945$
$b_2 = -0.64$

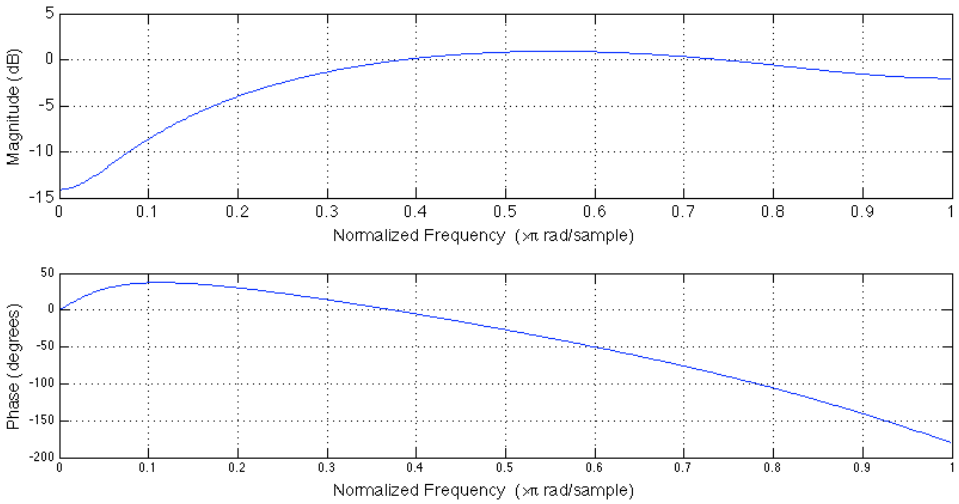

Figure A.1: Frequency and phase response of 2nd order pre-emphasis FIR.

Tanh[x]-mapping

$y[n] = \tanh(\alpha \cdot x[n])$, where $\alpha = 0.45$ was used in the experiments.

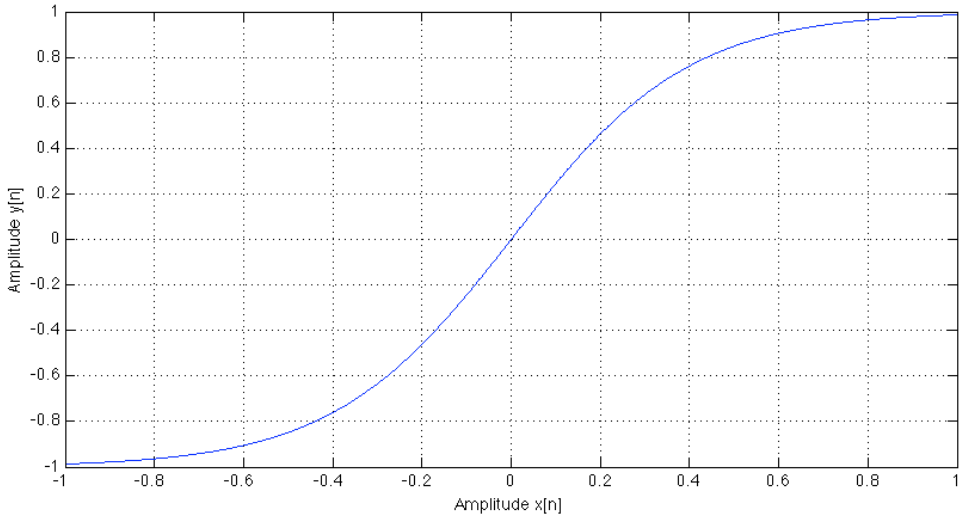

Figure A.2: tanh[x]-mapping of the spectral coefficients.

Minmax-filter for MATLAB

```
% Copyright © Okko Räsänen, 2007
%
% variables:
% s = s[m]
% I = index to the minimum
% mfiltlen = length of the filter

k = 1;
y = zeros(length(s),1);
while(k < length(s)-1-mfiltlen)
   nmm = s(k:k+mfiltlen-1);
   [A,I] = min(nmm);
   y(k+I-1) = (max(nmm)-A);
   k = k + 1;
end
```

Clustering algorithm fundamentals, pseudo-code

```
% fv = feature vectors for all segments
% X = cluster centroids for cluster space

while(k <= length(fv))

    for i=1:number_of_clusters
        dist(i) = calc_Dist(fv(k),X(i));     % Calculate distance
        [minValue,cIndex] = min(dist);       % Find index for minimum
    end
                                             % Check criteria
    if(minValue < tm && criteria1 == clusterCriteria1)
        merge(fv(k),X(cIndex));              % Criteria met, merge to existing
    else
        createCluster(fv(k))                 % Criteria not met, create new
                                               centroid
    end

    k = k+1;
end
```

Memory requirements of the clustering algorithm

A simple calculation demonstrates the memory requirements of clustering if every vector becomes stored separately. With a normal combination of onset and offset parts of the segment, the feature vector consists of 105 elements. In TIMIT there are on average 12 segments per second, and speech signals are on average approximately 2.5 seconds long, summing up to a total 30 segments per utterance. For example, in the TIMIT train/male section there are 3260 sentences, which results in almost 100k segments. Calculating 100000 * 105 = 10500000, or, 10.5 million elements, each requiring 8 bytes of memory. This means that merely storing the train-set segment feature vectors would require almost 100 megabytes of memory only for about two hours of speech. In the case of another database, TI-digits, there are almost 25k utterances total with an average of 15 segments per utterance, resulting in more than 300 megabytes of data. While this is not an exceptionally impossible task for modern computers, problems will rise when more and more material is fed to the system. On the other hand, if we store only cluster centroids separately and we end up with 500 clusters for the material, we only need 500*105*8 = 420 000 bytes for the entire cluster space. Adding new speech material will not necessarily increase the number of clusters, as it will already converge to some value depending on the distance criteria.