Chapter 1

Introduction

The process of communication between humans is mainly based on the ability to recognise and understand the speech signals transferred between them. Speech forms an integral part of the way humans interact with each other, as it is a highly effective and efficient way of exchanging information.

The automatic recognition of human speech by machine is regarded as a particularly difficult problem [1]. After decades of research, the goal of recognition of fluent, spontaneous speech, and the comprehension of its meaning, from any speaker in any environment, is far from being realised. The main success of speech recognition is due to the realisation that machines are not yet able to reach the performance of humans, and consequently, applying the technology only in a constrained way. This includes the use of only one speaker instead of many speakers (speaker dependent vs. speaker independent systems), the use of a small instead of a large vocabulary, and a well structured dialogue between man and machine, instead of an entirely open conversation. By realising the limitations of the current technology, and applying it only in an application specific way, speech recognition systems can be built that have acceptable performance.

Most speech recognition systems use some form of parametric model. The parame-

ters of these models are usually estimated from a database of training data during a training phase. After training the recogniser, speech can be recognised, using the trained models. The recognition phase thus refers to the process of recognising which models correspond to different parts of the speech signal. Usually the models of the recogniser correspond to some subword unit, such as a phoneme, since the models for these subword units can be reliably estimated with a limited amount of data.

Speech is usually recorded and stored on a sentence by sentence basis. The purpose of speech segmentation is to determine the boundaries of the recognition units, so that each model of the recogniser can be trained with the correct segment of speech corresponding to the model. Speech segmentation also finds use in segmental speech recognition systems, where speech is first segmented, after which the segments are classified. Segmentation can thus be used only during the training phase, or during both the training and recognition phase. Other uses of speech segmentation include the determination of sentence boundaries, for the automatic creation of speech sentences (e.g. for the creation of speech databases) from continuous speech, such as broadcast audio, determination of word boundaries, improving recognition performance, etc.

The task of speech segmentation is also of critical importance for speech synthesis. Most successful speech synthesis systems today typically employ the use of segment concatination of speech units (i.e. phonemes, diphones, syllables, etc.) from input training corpora of between 1 to 10 hours of speech. More natural speech synthesis is possible if effective segmentation can be performed to extract reliable synthesis units [2].

Speech segmentation algorithms can be broadly classified as belonging to one of two categories, namely those that make use of the underlying sequence of recognition units (i.e. forced alignment), and those that do not. In the first case only the boundaries need to be determined for a fixed number of specified units. In the latter case, the number of recognition units, as well as where the boundaries occur between them in time, are unknown.

This dissertation presents a recurrent neural network segmentation system, capable

of segmenting speech into phonemes, in a speaker independent manner. The system does not make use of the underlying sequence of phonemes, as such a sequence is not always available or reliable (as in conversational speech). This is especially true for the phoneme recognition experiments conducted here, as the sequence of phonemes is unknown prior to the recognition process. It will also be shown how the locations of the phoneme boundaries are incorporated into the speech recogniser in a novel way, in order to improve the recognition performance of the baseline system.

1.1 Problem statement

A need exists for the reliable, automatic determination of speech subword unit boundaries. The incorporation of the information from the boundary locations into a baseline recogniser also needs to be investigated, as this could potentially improve recognition performance. The research given here thus aims to meet the following objectives, namely to

- provide a general system capable of segmenting pre-recorded speech signals in a speaker-independent manner, where the desired result is the location of phoneme boundaries (at the frame level),
- establish a baseline phoneme recognition system against which the methods developed here can be tested,
- incorporate the information of phoneme boundary locations into a phoneme recogniser, in order to improve the phoneme recognition performance, and
- develop a new technique of incorporating segmentation information in a phoneme recogniser.

In order to make the research viable, a number of assumptions must be made, including

- the use of an available, general purpose speech recognition system from Cambridge University, called the hidden Markov toolkit (HTK) [3], which will be used for all recognition experiments,
- evaluation of the methods only on American English (the TIMIT database), which reflects read speech and not spontaneous speech, and
- processing of speech only in an off-line manner (i.e. where all of the speech is available at all times).

1.2 Summary of related work

This section presents a concise literature survey of concepts related to those used in this dissertation. For a complete review of speech recognition theory, the book by Rabiner and Juang [1] is recommended.

The comparison of different speech segmentation and recognition algorithms is a difficult task. Researchers tend to use different databases and performance measures. Only results on the TIMIT database are thus given here, unless other results can provide further insight into a particular algorithm or technique.

The TIMIT database [4] was designed to provide speech data for the acquisition of acoustic-phonetic knowledge. It is also used for the development and evaluation of automatic speech recognition systems. The speech was recorded at Texas Instruments (TI), transcribed at the Massachusetts Institute of Technology (MIT), and maintained, verified, and prepared for CD-ROM production by the US National Institute of Standards and Technology (NIST). It is currently available from the Linguistic Data Consortium (LDC). This database contains a total of 6300 spoken sentences, where 630 speakers each spoke a total of 10 sentences. The 10 sentences are made up of 2 dialect sentences (SA), 5 phonetically compact sentences (SX) and 3 phonetically diverse sentences (SI). For all our experiments, the SA sentences were ignored. This resulted in

3696 training files and 1344 test files (from which a subset of 192 sentences makes up the core test set).

The concepts presented in this dissertation make use of three parts, namely speech signal processing, segmentation, and recognition. Each of these components is discussed in the following sections in more detail.

1.2.1 Speech signal processing

Speech signal processing is usually the first step in segmenting or recognising speech. The main aim of this stage is to provide features which are better suited to the segmentation or recognition process than the raw data. The total number of these features is typically much smaller than the total number of raw speech samples.

Many different signal processing methods exist. General speech signal processing methods are discussed in [5]. Mel frequency cepstrum coefficients (MFCCs) are discussed in [6], as well as generalised MFCCs. Linear prediction coefficients (LPC) are discussed in [7], while auditory nerve representation is discussed in [8], and the bandpass liftering of speech in [9]. Vector quantisation (VQ), usually used in discrete HMM systems, is discussed in [10] and [11]. An algorithm to estimate the fundamental frequency of speech is given in [12]. The following paragraphs give the specific features used in the segmentation methods of the next section.

Vorstermans *et al.* [13] used an auditory model that incorporates an auditory filter bank, a bank of hair-cell models that emphasised the transitions at the phonetic boundaries, and a bank of envelope detectors that measured the envelopes of the hair-cell outputs in the different channels of the model. An acoustic vector was constructed every 10 ms, that contained an auditory spectrum (20 channels), difference spectrum, voicing evidence, a fundamental frequency (if enough voicing evidence was present), and energy from an energy function sampled at multiples of 2 ms (obtained by accumulating the hair-cell output envelopes across the different channels).

Pauws *et al.* [14] used five measurements that took into account the fact that unvoiced sounds are generally characterised by an energy concentration in the relatively high frequency region, while voiced sounds have an energy concentration in the lower frequencies. Speech and silence were distinguished through the use of an energy level measurement. The five measurements thus included the short-time energy, normalised such that silence was characterised by a value close to 1, normalised low-frequency energy in the range of 50 to 1200 Hz, normalised high-frequency energy in the range 2000 to 4000 Hz, the zero crossing rate, and the first linear predictive coefficient of a first-order LPC model. The 16 filterbank values, their first and second derivatives, and energy were also used. Pre-emphasis was used, with a coefficient of 0.95, as well as a Hamming window of 20 ms length with frame shifts of 2.5 ms, 5 ms, and 10 ms.

Bonafonte *et al.* [15] calculated an acoustic vector every 10 ms, by analysing speech frames of 20 ms using a Hamming window. For each frame, 20 mel-scaled filters were transformed to 12 MFCCs and a measure of the power, as well as their first and second derivatives were calculated. Only the 12 MFCCs were used to refine the boundary positions.

Olsen [16] used 31 triangular filters spaced linearly along the logarithmic mel scale, where each filter overlapped 50% with its two neighbours. Normalised log energy was also used. All the feature vectors were obtained from a 25.6 ms Hamming window with a 10 ms frame period. A total of 15 MFCCs, normalised log energy, and their first and second order derivatives, were used.

Lee [17] used 14 MFCCs with log energy, computed at 5 ms intervals. Pellom and Hansen [2, 18] parameterised the speech waveform every 5 ms, by a vector consisting of 12 MFCCs and normalised log-frame energy, as well as their first derivatives.

Jeong and Jeong [19] used a 256 point rectangular window, spaced at half a window size, and the LPC Burg algorithm for cepstrum analysis. The acoustic vector was then constructed from 16 whitened LPC cepstrum coefficients, obtained by using the whitening method. Policker and Geva [20] also used 12 LPC coefficients.

Cosi [21, 22, 23] allowed the use of many different kinds of features in an interactive segmentation and labelling automatic module (SLAM). The joint synchrony / mean-rate (S/M-R) model of auditory speech processing (ASP), fast Fourier transform (FFT) cepstrum, LPC based spectrograms, energy, pitch, and zero crossing, are some of these. The use of auditory model (AM) techniques was strongly supported.

Smith [24, 25] used an auditory front end, which used a Gammatone filter bank that bandpassed the signal into a number of channels. These were then rectified to model the effect of a set of inner hair cells.

Chang *et al.* [26] calculated a feature vector every 10 ms after several stages of processing. The first step was to compute a power spectrum every 10 ms over a 25 ms window. The power spectrum was then partitioned into quarter-octave channels between 0.3 and 4 kHz and logarithmically compressed in order to preserve the general shape of the spectrum distributed across frequency and time.

Regel [27] used normalised energy (frequency range 100 to 900 Hz), logarithm of the normalised energy, normalised autocorrelation coefficient at unit delay, first linear prediction (LP) coefficient, logarithm of the normalised LP error, and normalised amplitude, frequency, and bandwidth of the absolute maximum in the spectrum, to classify speech into one of a few broad categories ("silence", "voiceless", "voiced fricative", and "voiced non-fricative"). For classification of frames into the phone components, knowledge of the first classification stage allowed special features to be used for each of the broad categories. For the category "voiced non-fricative", normalised energy, logarithm of the normalised energy (frequency range 640 to 2800 Hz), normalised autocorrelation coefficient at unit delay, and normalised amplitude, frequency, and bandwidth of the lowest three formants were used. For voiceless sounds, the logarithm of the normalised LP error, normalised energy, logarithm of the normalised energy in five non-overlapping frequency ranges, normalised autocorrelation coefficient at unit delay, and normalised amplitude, frequency, and bandwidth of the absolute maximum in the spectrum were used. For the voiced fricative sounds, the same features were used as for voiceless

sounds, except for normalised amplitude, frequency, and bandwidth of the absolute maximum in the spectrum.

Fukada *et al.* [28] calculated an acoustic vector every 10 ms. The acoustic vector included the 12 MFCCs, power, and the first derivatives of these. A window of 25.6 ms was used.

In conclusion, it can be seen that many different speech features have been used in the past. MFCCs and energy have however proven to be the choice in recent years, not only for segmentation but also the recognition of speech. In the work presented here, the same MFCC and energy features will thus be used for segmentation and recognition tasks.

1.2.2 Speech segmentation

Linguistically constrained (explicit) segmentation

When the underlying sequence of phones is known, the segmentation algorithm only has to calculate the location in time of the boundaries between the phones (i.e. referred to as "forced alignment"). These methods perform reasonably well, as the higher level of lexical information (phoneme sequence) is used in the segmentation process. It is important to note that in most cases text information is provided, so reliable word to phoneme sequence look-up is necessary. This also assumes that speech production of the word set occurs without alternative pronounciations, otherwise the segmenter must consider alternative pronounciations during processing. The following is a short summary of some of these methods.

Vorstermans *et al.* [13] developed a system for the automatic segmentation and labelling of speech. The system first did initial segmentation by identifying major changes (landmarks) in the acoustic signal obtained from an auditory model. This was achieved by a

landmark identification, generation and elimination stage. Up to 4 consecutive initial segments were then merged to construct a set of candidate phonetic segments. A multilayer perceptron (MLP) was subsequently used in a phonetic segmentation stage to compute the probability of a boundary being a phonetic boundary, given the evidence of a preceding phonetic boundary and acoustic evidence. Phonetic classification was then performed through the use of another MLP that classified the acoustic vector into one of 5 broad phonetic classes, where the MLP's outputs were also interpreted as probabilities. A Viterbi search procedure aligned the speech with a state-transition model, derived from the transcription of the utterance, by taking the outputs of the two MLPs in the phonetic segmentation and classification stages into consideration. The result of the Viterbi search was a set of boundaries and labels, that maximised the combined likelihood of the phonetic boundaries and phone sequence, given the provided transcription and acoustic observations. The system was easily adaptable from one language to another, without the requirement of extensive linguistic knowledge or large (manually segmented and labeled) training databases of that language. A correct boundary placement of 76% (within 20 ms of the desired boundary location) was obtained on the core test set of the TIMIT database (5 SX sentences per speaker), with 48 phone labels. The speech was aligned against the manual transcriptions (only using the labels, not the manually provided boundary locations) of the TIMIT database, after adapting the baseline Flemish system, using 100 sentences of the TIMIT training set. A further gain of 5% on the overall system performance was achieved by also using the manually provided boundary locations, with about 200 training sentences, but it was not stated what the gain in segmentation accuracy was.

Pauws *et al.* [14] made use of the time alignment of the speech waveform against a sequence of HMMs, where each HMM represented a phoneme-like unit in the phonetic transcription of the utterance. Initialisation of the HMMs was performed by a 3-stage hierarchical procedure. The first stage involved the segmentation into broad phonetic classes (voiced, unvoiced and silence), on the basis of the phonetic transcription alone. This provided robust anchor points for the second stage, namely sequence-constrained vector quantisation (SCVQ), where the broad phonetic class regions were further de-

composed into their constituent phoneme-like units. Finally Baum-Welch estimation was used to fine-tune the HMMs. Segmentation was then performed by the Viterbi alignment of the utterances with the HMMs. An accuracy of 89.51% was obtained for a 20 ms tolerance, and 95.37% for a 30 ms tolerance, on a database consisting of 827 isolated words of the Dutch language. Learning was performed on the database to be segmented.

Bonafonte *et al.* [15] used hidden Markov models and the Viterbi algorithm to obtain an initial segmentation of the speech. A corrective procedure was then applied, which considered the segments of the segmented speech as homogeneous regions. A model was estimated for each segment of the utterance and Gaussian probability density functions (PDFs) were used to model the feature vector. Hypotheses for moving the boundary one frame to the left or to the right were then analysed. Boundaries were iteratively moved until no further changes occur. The result was that the boundary positions were refined and segmentation error was significantly decreased. They obtained an accuracy of 64.4% with a 12 ms window, and 81.3% with a 20 ms window, on the TIMIT database.

Olsen [16] also used hidden Markov models and Viterbi decoding of the speech utterance to segment the utterance, as well as Lee [17]. They did not report any specific segmentation results, and thus none of their results are given here.

Pellom and Hansen [2, 18] used dynamic programming (DP) to investigate the effect of different signal processing methods on the segmentation accuracy. Here the effect of noise on the segmentation performance was also investigated. They achieved segmentation accuracies of 47.9%, 69.9%, 85.9%, 95.9% and 98.4% for tolerances of less than 5 ms, 10 ms, 20 ms, 40 ms, and 60 ms, respectively, on the TIMIT database.

Jeong and Jeong [19] used a higher order Markov process, and the mean field solution to the segmentation problem, in a closed loop system consisting of combined bottom-up (segmentation, recognition and labelling) and top-down (labelling, speech generation and segmentation) processing. A recursive procedure provided an estimation of the

segmentation and phone label. Their system transformed the incoming continuous signal into one of the 61 phone classes at the rate of 73.7% when TIMIT was used.

Cosi [21, 22, 23] developed an interactive segmentation and labelling automatic module (SLAM). A multi-level segmentation theory was used. Speech was considered as a temporal sequence of quasi-stationary acoustic segments, where similarity of points in a segment is greater than for those between different segments. The segmentation problem was thus reduced to a clustering problem, where a decision was taken based on the similarity between the signal immediately preceding and following it. Initial "seed regions", which constitute the basis for the "hierarchical structuring", were created by a recursive technique that used a Euclidean similarity measure. Adjacent regions were then merged and a dendrogram was constructed. Pattern recognition techniques found the optimal segmentation path given the dendrogram structure and the target phonemic transcription.

Linguistically unconstrained (implicit) segmentation

When the underlying sequence of phonemes is unknown, the segmentation algorithm must not only estimate the location in time of the boundaries between phonemes, but also the number of boundaries (or alternatively the number of phones). These methods generally perform worse than those of the previous section, but are more versatile. The following is a short summary of some of these methods.

Smith [24, 25] used a general sound segmentation system to segment speech into phonemes. The sound signal was bandpass-filtered into a number of channels, rectified to model the effect of a set of inner hair cells, and filtered using an onset/offset filter. This made the transformed representation sensitive to energy rises and falls. The next step was to divide the onset/offset representation into two positive-going signals, an onset signal and offset signal. Both of these signals were then logarithmically compressed to increase the dynamical range of the system. These signals were sharp-

ened with an integrate-and-fire neural network, where the data was integrated across frequency bands and across time. The effect of this was to produce sharp onset firing responses across adjacent channels in response to a sudden increase in energy in some channels, thus grouping onsets both tonotopically and temporally. The outputs of the neural network sharpening stage were the onset and offset maps. The onsets were used for segmentation as the offsets tended to be more gradual and the continuous signal was divided at each onset. A minimum segment length of 25 ms was used, and the sharpness of the segmentation was varied by setting the minimum number of onset (offset) spikes which had to occur in the 10 ms window before that onset or offset line was taken to signal a segment start (end). Two male and two female sentences from each of the 8 dialect regions of TIMIT were segmented using this method. An average of 59% of the phoneme boundaries were correctly found (the estimated boundary and true boundary of a phoneme was within 15 ms of each other).

Chang *et al.* [26] used an array of independent, temporal flow neural networks that classified each frame into one of five articulatory-based phonetic-feature classes, namely place, manner of articulation, voicing, lip-rounding, and front-back articulation (for vocalic segments). They used a separate class for silence. These phonetic-feature labels were then combined and used as the input to an MLP network that gave a preliminary phonetic label to a frame. The last stage was a Viterbi-like decoding process that produced a sequence of phonetic-segment labels along with the times of the boundaries between them. They achieved 38.4%, 76.0% and 83.7% hits, and 58.5%, 20.9% and 13.2% false alarms, for a frame tolerance of 10 ms, 20 ms and 30 ms, respectively on the Oregon Graduate Institute (OGI) Numbers95 corpus.

Regel [27] used two classification stages in an acoustic-phonetic transcription system. In the first stage a decision was made in favour of one of four categories, namely "silence", "voiceless", "voiced fricative", and "voiced non-fricative". A Bayes classifier was used for this purpose. The second stage consisted of the classification of the frames into phone components, using the results of the first step. In this stage only special features were used for each class. A Bayes classifier was also used for this purpose. The resultant

probabilities of the two stages were then multiplied together to obtain an estimate of the *a posteriori* probability of the phone component. Adjacent frames were then combined in a two-stage process. In the first step, similar frames were lined up with simultaneous smoothing. In the second step, essential segments were extracted and attempts made to fill the gap between two essential segments, using a similarity measurement. The results given are not directly comparable to the work presented here, and are thus not given.

Policker and Geva [20] regarded the speech signal as a non-stationary time series. They developed a model and a set of algorithms to estimate the parameters of the non-stationary time series. Fuzzy clustering methods were used to estimate the continuous drift in the time series distribution and to interpret the resulting temporal membership matrix as weights in a time varying, mixture probability distribution function. A decision rule was imposed on the distribution temporal change, where a limiting procedure of any cluster crossing the 0.5 probability level, was used. This resulted in segmentation of the speech signal into phonemes. No specific results were given.

Andre-Obrecht [29] (also in Basseville and Nikiforov [30]) used a statistical approach. [1] The signal was modeled by an autoregressive (AR) statistical model. Test statistics were then used to sequentially detect changes in the parameters of the model. Three different segmentation algorithms were presented, differing in the assumption of the excitation of the model (glottal impulses), and the choice of the test statistics (generalised likelihood, or statistics of cumulative sum type). The segmentation was performed on a sample-by-sample basis, and not a frame-by-frame basis, allowing more accurate location of boundaries, and the possibility of using shorter segments. Their results are not directly comparable to the work presented here, and are thus not given.

A Bayesian autoregressive changepoint detector (BCD) was used by Èmejla and Sovka [31]. A three-step algorithm was used to segment the speech. In the first step, a segmenta-

[1] Examples of some of these methods can also be found on the Internet at http://www.cnmat.berkeley.edu/~ tristan/Thesis/timedomain.html

tion point was assigned to the centre of sound units composed of vowels and semivowels. This was done because the BCD was highly sensitive to spectral changes and was rather used to refine the positions gained from the first step. By using segmentation points inside stationary parts of the signal, the autoregressive order to the left and right sides of the data segment could be estimated with higher accuracy. The second step used a BCD between each pair of given segmentation points in the stationary parts of the speech. The final step used a BCD with data between the stationary segmentation point and the segmentation point gained in the second iteration. This method was suitable for vocal-consonant-vocal (VCV) structured utterances. No comparable results to the work presented here were given.

Petek *et al.* [32] investigated the robust automatic segmentation of spontaneous speech. They used the spectral variation function (SVF), which was defined as a correlation measure between successive windows of acoustic observation vectors, to segment the speech. They compared mel-frequency cepstra (MFC), relative spectral processing (RASTA), and forward-backward auditory masking dynamic cepstra (FBDYN) based SVF algorithms. The FBDYN-SVF method resulted in smoothing of the cepstra by the forward and backward masking lifter, giving an improvement over the other two methods. Their numerical results are not comparable to the work presented here, and are thus not given.

Fukada *et al.* [28] used a bi-directional recurrent neural network (BRNN) to segment speech. The system was trained to segment the speech signal into phonemes, using a target value of 1 for a frame in which a boundary occurred, and a target value of 0.5 for the frames to the left and right of the boundary frame. Frames in which no boundary occurred were given the target value of 0. The neural network estimated the probability of a boundary, given the acoustic vector. By using thresholds or a segment lattice, the segmentation points in the speech signal could be found. They found that the BRNN segmented the TIMIT database with 8.33%, 76.01%, and 79.61% accuracy for frame margins of 0, 1, and 2, respectively. They also found that normal MLP neural networks performed worse. An MLP with 1 context frame segmented the TIMIT database with

1.46%, 61.90%, and 68.64% accuracy, an MLP with 3 context frames 6.12%, 64.16%, and 70.94%, and an MLP with 5 context frames 6.20%, 64.69%, and 71.64% for frame margins of 0, 1, and 2, respectively.

Related segmentation tasks

This section briefly highlights research done in related segmentation tasks. These tasks perform segmentation at a level higher than the phoneme level, or in a limited way.

At the subword level, a number of other segmentation attempts are worth mentioning. Chan and Ng [33] discussed the separation of fricatives from aspirated plosives by means of temporal spectral variation. Ryeu and Chung [34] used chaotic recurrent neural networks (CRNN) to classify and segment Korean monosyllables. De Mori and Laface [35] used fuzzy algorithms to segment speech into vowel-consonant-vowel (VCV) pseudosyllables (PSS). In Shyu *et al.* [36], an automatic co-articulation segmentation algorithm was developed, that took co-articulation into account. Co-articulation was also taken into account by Yu and Oh [37], where a neural network (NN) was used to segment speech into non-uniform units. Co-articulation information and neural networks were also used by Hosom and Cole [38], where the neural networks segmented speech into diphones. Steady-state zones of all phones carrying a diphone boundary were specified by a centroid vector, and together with an objective distance measure, hypothetical boundary cost functions were used to extract diphone elements in Kaeslin [39]. Temporal flow neural networks were used by Shastri *et al.* [40] for finding the temporal boundaries of syllabic units. Hsieh *et al.* [41] also segmented speech into syllables, using a hybrid neuro-fuzzy network. Cook and Robinson [42] used an MLP to determine the onset of syllables.

The segmentation of speech into voiced, unvoiced, silence, and/or mixed regions of speech can also be done. Examples include [43, 44, 45, 46, 47, 48, 49, 50, 51].

Segmentation can also be performed at the word level. Zelinski and Class [52] seg-

mented an utterance into single words, using statistical principles. Christiansen *et al.* [53, 54] used a simple recurrent network (SRN) for this purpose.

At the sentence level, Siegler *et al.* [55] used the Kullback Leibler (KL) distance metric to segment broadcast news audio into discrete utterances. Prosody-based automatic segmentation of speech into sentences and topics was investigated by Shriberg *et al.* [56]. Semantic dialogue unit (SDU) segmentation, which correspond roughly to speech act (utterance) segmentation, was provided by a multi-level segmentation algorithm in Lavie *et al.* [57]. Speech act detection was also performed by Ries [58], using HMM and neural network based methods. Swerts and Ostendorf [59] investigated prosodic and lexical indications of discourse structure and utterance purpose. Tzanetakis and Cook [60, 61] developed a framework for audio analysis based on classification and temporal segmentation.

Segmentation at even higher level is also possible. A syntactic-prosodic labelling scheme was developed in Batliner *et al.* [62] that could segment speech into sentences or phrases. Renals *et al.* [63] developed a system that could segment speech into stories, for the indexing and retrieval of broadcast news. Speaker-based segmentation system for audio data indexing was performed by Delacourt and Wellekens [64]. Energy-based speech endpoint detectors were compared in Bush *et al.* [65].

1.2.3 Speech recognition

This section does not focus on background of hidden Markov models, neural networks, fuzzy logic, or other methods as general techniques. Instead, a very brief literature survey of the application of these methods to speech recognition, is presented.

For a detailed review of hidden Markov models, see [1, 66, 67, 68]. For neural networks, [69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80] can be recommended. Fuzzy logic was also used frequently and background on the underlying theory and techniques can be found in [81] and [82]. In addition to fuzzy logic, [83] also explains neuro-fuzzy concepts.

Artificial intelligence concepts are explained in [84]. Numerical methods used can be found in [85, 86, 87]. Statistical theory can be found in [88] and [89], while general pattern recognition concepts can be found in [90].

Hidden Markov models have proven to be very well suited to recognising human speech. Some of the main advantages of HMMs are that they integrate well into systems incorporating both task syntax and semantics. Examples of HMM-based speech recognition include [1, 66, 67, 91, 92, 93, 94, 95, 96]. Related to hidden Markov models is the concept of dynamic programming, such as [97] and [98]. Dynamic time warping (DTW), as well as a probabilistic matching algorithm was used by [99]. Similar concepts, some involving template matching, were also used by [100, 101, 102]. Fuzzy logic concepts were used by [103] and [104]. In using hidden Markov models, a number of assumptions are made. These include the assumptions that the speech signal can be well characterised as a parametric random process, that the parameters can be estimated in a precise, well-defined manner, and the fact that a first order Markov chain is usually used ([105] showed how higher-order HMMs can be used efficiently). Conventional HMM systems make use of an independence assumption of the observation.

Neural networks have also found their way into the area of speech recognition. This is partially due to the thin biological connection that exists between neural networks and the human brain, and the fact that neural networks operate well as pattern classifiers and can estimate probabilities conveniently. Time delay neural networks were used in [106] and [107]. Neural-fuzzy concepts were combined with an HMM-based automatic speech recognition system in [108]. Spiking neural networks were used in [109] and [110]. Simple recurrent neural networks, also called Elman neural networks, were used in [111] and [112] to discover symantic/semantic features of words, and in [113] was used for speech recognition, where the neural networks were trained with the leap-frog algorithm. Recurrent neural networks were used in [114, 115, 116, 117]. In [118] various different neural networks are discussed for use in the context of speech recognition. An advantage of neural networks for speech applications is that they are general and do not impose a rigid structure into the recognition process. Some weaknesses include the

increased size of training material over traditional HMMs, and their inability to allow for efficient adaption to changing noise conditions versus methods seen in HMMs. See [119] and [120] for more detailed information on the use of neural networks in speech recognition.

It is often the case that the best systems are hybrid ones. In speech recognition, hybrid systems perform particularly well. In [120] it is discussed how neural networks might be incorporated into HMM systems. Usually neural networks are used to estimate the observation density in the HMM states, or the *a posteriori* probability of a certain phone, given acoustic evidence. In [121] it was shown how the *a posteriori* probability of a complete utterance could be estimated, as an alternative approach to the regular split into acoustic model and language model likelihood. A bi-directional recurrent neural network estimated the occurring probability terms. In [42] syllable boundary information was included to improve the recognition process. Their method made use of two models for each phone, one model when the phone occurs at a syllable onset, and one when it does not. In [28] the transition probabilities of the HMMs were modified by a BRNN output. They also showed how the neural network could be used with a polynomial segment model (PSM) based recogniser. PSM based recognition systems do not rely on the observation independence assumption of conventional HMM systems. The neural network was used to segment the utterance into segments, where the boundary locations were determined by a segment lattice as a postprocessor.

1.3 Approach and research hypotheses

The work presented here addresses two types of problems, namely speech recognition and speech segmentation. It is also shown how information from the independent segmentation stage can be used to improve speech recognition.

Figure 1.1 gives an overview of the approach followed here. It consists of two independent components, namely a speech recognition component, and a speech segmentation

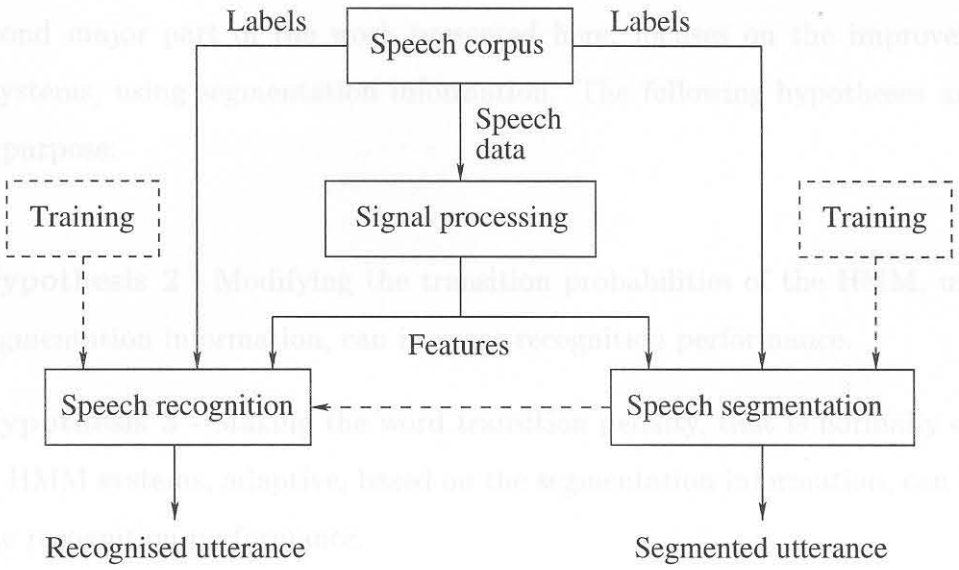

Figure 1.1: Overview of the approach taken.

component. A speech corpus contains label files and speech data files from a large number of speakers. These are used to make the system speaker independent. The signal processing module extracts features from the speech data that are useful for the recognition and segmentation stages. For simplicity, the recognition and segmentation stages presented here use the same set of features. Initially the recognition and segmentation components are trained independently using the corpus, and evaluated separately. Segmentation information is then used to improve the recognition performance of the entire system. American English (the TIMIT database) is used as the speech database.

As mentioned, one of the problems investigated here, is the problem of speech segmentation. It is known that hidden Markov model systems perform segmentation automatically as part of the Viterbi decoding process. Neural networks can also be used for the segmentation process. This leads to the following hypothesis:

- **Hypothesis 1** - Recurrent neural networks will perform the segmentation task (into phonemes) better than HMMs.

The second major part of the work presented here, focuses on the improvement of HMM systems, using segmentation information. The following hypotheses are made for this purpose:

- **Hypothesis 2** - Modifying the transition probabilities of the HMM, using the segmentation information, can increase recognition performance.
- **Hypothesis 3** - Making the word transition penalty, that is normally constant in HMM systems, adaptive, based on the segmentation information, can increase the recognition performance.

1.4 Contributions of this study

The work presented here offers a number of contributions. Not only is a new technique presented, but both old and new techniques are evaluated using American English (TIMIT) speech. Before this dissertation, the use of segmentation information in the recognition process of a standard speech recognition system, was not commonly used. The contributions of this dissertation include

- a high performance speech segmentation system, involving the use of a recurrent neural network, capable of segmenting speech into phonemes, without the use of higher-level lexical knowledge,
- a new technique of incorporating segmentation probabilities into the speech recognition system, in order to improve phoneme recognition performance,and
- evaluation of the methods, of incorporating segmentation information into HTK, on the TIMIT database, is presented in order to see how well they perform in a state-of-the-art speech recognition system.

It is shown that the new technique developed here outperforms some techniques used by others. It is also shown that the technique is fairly efficient and can be easily incorporated into a standard recogniser.

1.5 Dissertation outline

Chapter 2 presents the background theory relevant to the work presented here, in speech signal processing (2.1), hidden Markov models (2.2), and recurrent neural networks (2.3).

Chapter 3 provides detail about the segmentation process. Aspects discussed include preprocessing (3.1), segmentation (3.2), using hidden Markov models (3.2.1) and recurrent neural networks (3.2.2), postprocessing (3.3), and the accuracy measure used (3.4).

Chapter 4 deals with speech recognition. Baseline recognition (4.1), recognition using segmentation information (4.2), and the accuracy measure used (4.3), are discussed.

Chapter 5 gives the experimental results. In this chapter the concepts of Chapters 3 and 4 are evaluated, using the defined accuracy measures. Speech segmentation (5.1) methods include the use of hidden Markov models (5.1.1), as well as recurrent neural networks (5.1.2). Recognition experiments include the baseline system (5.2) and recognition using the segmentation information (5.3).

Chapter 6 gives a summary and some conclusions are made. A summary of results (6.1), statistical significance testing (6.2), conclusions (6.3) and shortcomings and future work are discussed (6.4).

Finally, Appendix A gives the details on training recurrent neural networks. The back-propagation through time (BPTT) technique is discussed in the context of recurrent neural networks, and it is shown how bi-directional recurrent NNs (BRNN) are trained.