Confidence Measures
for Large Vocabulary
Continuous Speech Recognition

Matt Seigel
Department of Engineering
University of Cambridge

First Year PhD Report
27 August 2010
Abstract

Large vocabulary speech recognition with multiple speakers and variable channel conditions is a challenging task. State-of-the-art Large Vocabulary Continuous Speech Recognition (LVCSR) systems are nevertheless able to generate high quality transcriptions of speech utterances under such conditions. These transcriptions are however not entirely accurate for a number of reasons. This fact has given rise to the field of Confidence Estimation (CE) for speech recognition systems, in which the aim is to investigate methods through which accurate measures of confidence in recognition results may be obtained. Such measures have become increasingly useful within many areas of speech recognition research, particularly those in which recogniser output is used as input for downstream applications.

This report aims to address the problem of CE in LVCSR within the domain of Conversational Telephone Speech (CTS). The fundamental premise of this work being that improved confidence measures may be obtained through the application of alternative modelling techniques and algorithms, which are able to leverage the available sources of information to greater effect. The current state-of-the-art is therefore assessed in some detail, before new approaches that may prove useful in yielding improved confidence measures are presented and preliminarily investigated.
Contents

Contents ii
List of Figures vi
Notation vii

1 Introduction 1
  1.1 Problem Statement .................................................. 2
  1.2 Approach .......................................................... 2
  1.3 Structure of the Report ............................................. 3

2 Automatic Speech Recognition 5
  2.1 Pattern Classification ............................................. 6
  2.2 Statistical Pattern Classification in ASR ....................... 6
    2.2.1 Feature Extraction ........................................... 6
    2.2.2 The Classification Task ................................... 7
  2.3 Acoustic Modelling ............................................... 8
  2.4 Language Modelling .............................................. 10
  2.5 Search .......................................................... 11
  2.6 Lattices .......................................................... 12
  2.7 Evaluation ....................................................... 13

3 Confidence Measures for Speech Recognition 15
  3.1 Classification Approach ......................................... 16
    3.1.1 Predictor Features ........................................... 16
    3.1.2 Shortcomings ................................................ 18
### 3.2 Posterior Probability Approach
- 3.2.1 Time-based Posteriors
- 3.2.2 Confusion Network Clustering
- 3.2.3 Shortcomings

### 3.3 Utterance Verification Approach
- 3.3.1 Likelihood Ratio Testing
- 3.3.2 Shortcomings

### 3.4 Evaluation of Confidence Measures

### 3.5 Comparison of Approaches
- 3.5.1 Information Sources
- 3.5.2 Complexity of the Approach
- 3.5.3 Flexibility in Score Granularity
- 3.5.4 Modelling Additional Information
- 3.5.5 In-system Application of Confidence Scores

### 4 Classifying Sequential Data
- 4.1 Models for Sequence Classification
  - 4.1.1 Generative Models
  - 4.1.2 Discriminative models
- 4.2 Linear-Chain CRFs
  - 4.2.1 Definition and Derivation
  - 4.2.2 Properties of Interest
  - 4.2.3 Feature Functions and Input Features
  - 4.2.4 Parameter Estimation
- 4.3 General CRFs
  - 4.3.1 Higher Order/Skip-chain CRFs
  - 4.3.2 Hidden-state CRFs
  - 4.3.3 Hierarchical CRFs

### 5 Confidence Estimation on Lattices
- 5.1 Direct Computation on a Lattice
  - 5.1.1 CE using Lattice-based MBR
- 5.2 Direct Classification on a Lattice with CRFs
## Contents

5.2.1 Input Features from the Lattice .................................. 50  
5.2.2 CRF-based Clustering .............................................. 51

6 Confidence Estimation with Rich Features ............................... 53  
6.1 A CRF-based Approach ................................................ 53  
6.1.1 Input Features .................................................... 54  
6.1.2 Complex Feature Functions ...................................... 55  
6.1.3 Hierarchical Representation ...................................... 56  
6.1.4 Phrase-level Measures of Confidence .............................. 57
6.2 LRT ................................................................. 59

7 Implementation of a CRF Toolkit ....................................... 60  
7.1 Data ................................................................. 61  
7.2 Feature Representation ................................................ 62  
7.2.1 Discrete Features .................................................. 62  
7.2.2 Continuous Features .............................................. 62  
7.2.2.1 Moment Constraints with Binning .......................... 63  
7.2.2.2 Distribution Constraints with Cubic Splines ............... 63  
7.3 Parameter Estimation ................................................ 65  
7.4 Scoring Transcriptions ............................................... 65  
7.4.1 Word Sequence Tagging ......................................... 66  
7.4.2 Word-level Confidence Measures ................................ 66

8 Experiments ..................................................................... 67  
8.1 Data Sets ............................................................... 67  
8.2 Lattice-Based Posterior Experiments .................................. 68  
8.2.1 Posterior Estimation ............................................... 68  
8.2.2 Baseline Definition ............................................... 68  
8.2.3 Baseline Evaluation ............................................... 69  
8.2.4 CRF Parameter Experiments ...................................... 70  
8.2.5 Discussion .......................................................... 71  
8.3 Alternative Input Features ............................................ 73  
8.3.1 The N-Best Homogeneity Score .................................. 73  
8.3.2 The LM Back-off Measure ........................................ 74
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>HMM Architecture</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>An Example Word Lattice</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>An Example Confusion Network</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>Sequential Model Architectures</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>HCRF Architecture</td>
<td>42</td>
</tr>
<tr>
<td>4.3</td>
<td>Hierarchical CRF Architecture</td>
<td>44</td>
</tr>
<tr>
<td>8.1</td>
<td>DET Curves for CNP Baselines and Parameter Experiments on Eval04</td>
<td>72</td>
</tr>
<tr>
<td>10.1</td>
<td>Proposed Work Schedule</td>
<td>91</td>
</tr>
</tbody>
</table>
Notation

Useful Definitions

\( \Lambda_w \) A model which represents a null hypothesis in the Utterance Verification framework.

\( \Lambda_a \) A model which represents an alternative hypothesis in the Utterance Verification framework.

\( f_k \) A generic CRF feature function.

\( Z(\cdot) \) A normalisation function.

\( \delta \) The Kronecker delta function.

\( \lambda_k \) The weight parameter for a generic CRF feature function \( f_k \).

\( \mathcal{L}(\cdot) \) The log likelihood function.

Mathematical Notation

\( x \) A discrete or continuous scalar value.

\( \mathbf{x} \) A vector.

\( X \) A random variable taking on values \( x \).

\( \mathbf{X} \) A general matrix.

\( P(X) \) The probability mass function of a discrete random variable \( X \).

\( p(X) \) The probability density function of a continuous random variable \( X \).
Chapter 1

Introduction

The task of automatically generating reliable transcriptions of audio containing speech is one which has attracted a great deal of research interest. State-of-the-art systems employ complex statistical models of speech and language in order to accomplish this task. Given enough data for a particular domain, these models can be reliably estimated with proven techniques to yield sophisticated Automatic Speech Recognition (ASR) systems. However, the migration of such systems from the laboratory environment to real-world applications invariably has a detrimental effect on performance. A number of factors contribute to this fact, the most prevalent of which are due to differences between the data used to design the system, and that encountered in the real world. Channel distortions, speaker variations and ambient noise are some examples of such factors commonly cited as sources of error. Furthermore, assumptions that are typically made when estimating the underlying statistical models are often too strong - or even incorrect. The effect these factors ultimately have is that the word-level transcriptions generated by the ASR system often contain a significant number of errors. It is therefore vitally important for a number of ASR applications, that such systems have information pertaining to the recogniser’s confidence in its transcriptions. These scores are typically defined at the word-level, as this is the most intuitive representation. The task of estimating such confidence measures for ASR systems is known as Confidence Estimation (CE).
1.1 Problem Statement

The broad scope of this work is to address the problem of improving CE for Large Vocabulary Continuous Speech Recognition (LVCSR) systems. The specific CE task at hand and underlying LVCSR system should however be specified in more detail.

This work will be primarily concerned with the estimation of word-level confidence scores. This is the CE approach most commonly taken in the literature. However, confidence measures defined at alternative levels of granularity will also be investigated. The definition of error regions described in [3], allows for a more general view of the granularity at which a confidence measure is defined. An error region is therefore considered here to be a span of contiguous linguistic units (i.e. phones, words etc.), which are to be considered as potentially errorful.

The LVCSR task domain within which the work described here will be carried out is that of Conversational Telephone Speech (CTS). ASR applications such as dialogue systems and voicemail transcription services stand to benefit significantly from improved confidence measures, and are both essentially CTS tasks. This domain is therefore one of particular interest for research into CE techniques.

The problem addressed may therefore be restated as that of improving the accuracy of confidence scores assigned to each suitably defined error region within the best transcription of a CTS utterance generated by an underlying LVCSR system.

1.2 Approach

Given the problem definition in Section 1.1, the approach to be taken in this work may be defined. The fundamental premise is that improved confidence measures may be obtained through the application of alternative modelling techniques and algorithms. It is hypothesised that these alternative models/techniques will prove more adept at leveraging the given information sources in estimating confidence measures. Discriminative models are of particular interest, and shall play a significant role in the statistical modelling aspect of the approach taken. Furthermore, the use of models which are suited to the sequential nature of the data (word
sequences), will be investigated.

The viewpoint of confidence estimation over error regions, as described in Section 1.1, will be assumed in this work. This view allows for the investigation of measures defined at various levels of granularity.

The information which is readily available within the ASR system is indicative of the operation of the recogniser. This information will therefore form the foundation of that used by the CE approaches described. However, the use of additional information sources such as alternative forms of statistical models to those used by the recogniser, will also be investigated.

1.3 Structure of the Report

The report is structured as follows:

- Chapter 2 provides a broad overview of ASR systems and introduces a number of themes which will be mentioned throughout this report.

- Chapter 3 serves as an account of the literature on CE for ASR systems. The strengths and weaknesses of the existing approaches will also be discussed and comparisons will be made between the approaches on this basis.

- Chapter 4 will introduce the problem of sequence classification. A number of competing models suited to this problem will briefly be presented, before a class of models of particular interest is discussed in some detail.

- Chapter 5 presents approaches for future work in CE which operate directly on recognition lattices. Lattice-based algorithms and a classification approach are discussed.

- Chapter 6 presents approaches for future work in CE which are based on alternative “rich” sources of information. Statistical frameworks under which these sources may be successfully combined are discussed in some detail.
Chapter 7 outlines the details of a statistical model toolkit that has been developed as part of this work, and will be used for a number of the experiments described in Sections Section 8 and Section 10.

Chapter 8 details a number of preliminary experiments which have been carried out, and provides some analysis of the results of these experiments.

Chapter 9 draws conclusions which impact future work based on the state-of-the-art as described in the literature survey (Section 3 and Section 4). Conclusions are also drawn on the results of the preliminary experiments presented in Chapter 8.

Chapter 10 presents a work plan for investigating the research questions set out in Section 5 and Section 6. A schedule for the workplan is also provided.
Chapter 2

Automatic Speech Recognition

Automatic Speech Recognition (ASR) is defined as the process of translating an acoustic speech waveform into a corresponding textual representation, or transcription. The complexity of a speech recognition task is normally expressed through the definition of a number of crucial application parameters, which are outlined below.

- The vocabulary size under consideration. Small vocabulary tasks are those in which the range of input is greatly constrained (such as digit/letter recognition tasks). Large vocabulary tasks are typically defined as those in which vocabulary sizes of over 10,000 words are considered.

- The speaking style. In isolated speech recognition tasks, it is assumed there are clearly distinguishable pauses between each word in the speech utterance. Conversely, in continuous speech recognition tasks, this assumption is relaxed to suit more “natural” speech with arbitrarily short pauses between words.

- Speaker variability. Speaker dependent ASR systems constrain the recognition task to one particular speaker, while speaker independent systems are designed to recognise speech produced by any arbitrary speaker. State-of-the-art ASR systems tend to employ speaker adaptation techniques [12] in order to adapt to different speakers dynamically.
The acoustic environment and channel conditions. The acoustic environment has a significant impact on the quality of the audio to be transcribed, as does the quality of the channel over which the audio is recorded. As such, ASR systems may be designed for operation in noisy environments with high levels of background noise, or where the channel is particularly noisy (e.g. telephone recordings).

Modern ASR systems are founded on the principles of signal processing, statistical pattern processing and statistical models of spoken language. The reliable estimation of such statistical models from the appropriate speech/text corpora is achieved through the application of suitable machine learning techniques. In the sections which follow, a more detailed treatment of the afore-mentioned principles which are particularly relevant in the context of this report will be presented.

2.1 Pattern Classification

In pattern classification, it is assumed that any data point (or pattern) of interest, may be classified as belonging to only one class \( \omega \) from a set of \( n \) classes \( \Omega = \omega_1, \ldots, \omega_n \). Each such pattern may be represented by a set of \( d \) informative measurements/values called features. These features for each pattern are typically combined into a feature vector, \( \mathbf{x} = (x_1, \ldots, x_d) \). Pattern classification may then be defined as the process of applying some mapping function \( f \) to a given feature vector \( \mathbf{x} \), such that it may be labelled as belonging to some class \( \omega \), or \( f : \mathbb{R}^d \rightarrow \Omega \). The parameters of this mapping function may be estimated from training data.

2.2 Statistical Pattern Classification in ASR

2.2.1 Feature Extraction

In theory, a raw discretised version of the speech waveform should be an adequate representation of the speech signal, and be useful in subsequent processing steps. This is however not the case. There is a great deal of variability in raw speech
signals, warranting the implementation of front-end processing techniques to aid in smoothing the audio spectrum before carrying out classification tasks. This process is known as feature extraction. Feature extraction from speech signals is carried out by initially segmenting the raw speech waveforms into frames which are roughly 10-20ms in length, over which the waveforms are assumed to be stationary. Spectral properties of each segment are then calculated in order to yield a low-dimensional representation of the speech segment \( \mathbf{x} \). These spectral properties are usually calculated based on some model of human hearing, as is the case in the Mel-scale and Perceptual Linear Prediction (PLP) techniques.

### 2.2.2 The Classification Task

Speech recognition may be formulated as a pattern classification task. If one considers the problem of isolated word recognition, the set of classes \( \Omega \) corresponds to the words in the recognition vocabulary \( \mathcal{V} \). The feature vector \( \mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_T \) corresponds to the sequence of observation feature vectors extracted from the \( T \) segments in the speech waveform. With the aim of such a system being to recognise a single word, the optimal classifier for this task may be defined as one which assigns a word label to observation vectors based on the following rule:

\[
\hat{\omega} = \arg\max_{\omega \in \mathcal{V}} P(\omega|\mathbf{X}),
\]

where the term \( P(\omega|\mathbf{X}) \) represents the a-posteriori probability of the class/word \( \omega \). This form of classifier will therefore assign the class \( \omega \) with the highest a-posteriori probability to the pattern \( \mathbf{X} \), and is referred to as a Maximum A-Posteriori (MAP) classifier. Equation (2.1) may be expanded with Bayes theorem \( (P(A|B) = \frac{P(B|A)P(A)}{P(B)}) \) to yield:

\[
\hat{\omega} = \arg\max_{\omega \in \mathcal{V}} \frac{p(\mathbf{X}|\omega)P(\omega)}{p(\mathbf{X})} \propto \arg\max_{\omega \in \mathcal{V}} p(\mathbf{X}|\omega)P(\omega).
\]

(2.2)

The denominator term \( p(\mathbf{X}) \) is generally omitted as it is independent of the class, and therefore has no significant bearing on the decision rule. Assuming the distributions \( p(\mathbf{X}|\omega) \) and \( P(\omega) \) are the “true” distributions, a classifier which
implements this decision rule is guaranteed to minimise the misclassification rate

1 [7].

Extending the formulae introduced thus far for isolated word recognition, to the continuous speech recognition case is trivial. Considering that the classifier should now assign sequences of words, $W$, to a feature matrix $X$ of observations, equation (2.2) may be restated as:

$$
\hat{W} = \arg\max_W \frac{p(X|W)P(W)}{p(X)} \propto \arg\max_W p(X|W)P(W). \quad (2.3)
$$

The probability distributions on the right hand side of equation (2.3) are unknown. They are therefore typically estimated from training data, and subsequently substituted into the decision rule formula. This estimation process is detailed in the following two sections. Despite the fact that the resulting model distributions are indeed approximations to the “true” distributions, the optimal classification rule holds under the assumptions of model correctness.

### 2.3 Acoustic Modelling

The class-conditional probability distribution $p(X|W)$, or Acoustic Model (AM), is typically estimated from a parallel training corpus of speech segments/utterances and the associated transcriptions. This is not a trivial task as the high-dimensional form of the observation matrix $X$ makes direct estimation of the conditional likelihood intractable. A common solution to this challenge is to model the joint probability $p(X,W)$, with a set of parametric models of word production having parameters $\theta$. Under this generative framework, it is assumed that the sequence of observation vectors for a given word could have been generated by a Markov model.

A Markov model is essentially a finite state machine with a number of states, each having a unique form of output distribution. At each time step $t$ a transition is made from a state $i$ to $j$, and an output symbol $o_t$ is generated from the output distribution $b_j(o_t)$ of the new state. The transitions are governed by the

---

*The misclassification rate is defined as the probability of assigning a feature vector to the wrong class, $P(\hat{\omega} \neq \omega)$. 

---
probabilities $a_{ij}$. Thus, for a given state sequence $q$ of length $T$ this yields the following:

$$p(X, q|\theta) = \prod_{t=0}^{T-1} a_{q(t)q(t+1)}b_{q(t)}(X_t),$$  \hspace{1cm} (2.4)

where the state sequence $q$ which generates a given output sequence is not known in practice. Such a model is known as a Hidden Markov Model (HMM), the structure of which is shown in Figure 2.1. The vertices in this figure are shaded to indicate that the corresponding variables are unobserved.

In order to obtain the desired observation likelihood, it is therefore necessary to marginalise out over all possibilities for the hidden state sequence $q$ as follows:

$$p(X|\theta) = \sum_q \prod_{t=0}^{T-1} a_{q(t)q(t+1)}b_{q(t)}(X_t).$$  \hspace{1cm} (2.5)

Given equation (2.5), and assuming that the distributions are independent, a solution to the modelling problem may be defined as follows:

$$p(X|W) = \prod_{\omega \in W} p(X|\theta_\omega),$$  \hspace{1cm} (2.6)

where $\theta_\omega$ are the parameters of the specific HMM for the word $\omega$. The state output distributions $b_j(\cdot)$ are typically modelled using Gaussian Mixture Models (GMMs). The parameters of the GMM must be estimated, along with the transition probabilities in order to yield the parameter set $\theta_\omega$. These parameters are chosen so as to maximise the likelihood of the model generating the training data (Maximum Likelihood (ML) estimation). Typically, the Baum-Welch algorithm which is an instance of the class of iterative Expectation Maximisation (EM)
algorithms, is used in order to estimate these parameters.

The treatment of HMM-based recognition given here assumed that each HMM represented linguistic units at the *word level*. However, in LVCSR the size of the vocabulary means that there is not enough training data to reliably estimate the models for many words. It is therefore more common in real-world ASR systems for each HMM to represent a sub-word unit such as a phoneme, for which there is an adequate amount of training data. These models may then be concatenated together based on a pronunciation dictionary to yield more accurate word-level representations.

A recent advance in state-of-the-art acoustic modelling is that of *discriminative training*. This approach is concerned with the estimation of parameters for an inherently *generative* model using *discriminative* training criteria. These approaches do not suffer as heavily from assumptions made in modelling with respect to the distribution of the data $p(X)$, as the joint distribution is no longer modelled. Discriminative criteria such as Minimum Classification Error (MCE) and Minimum Phone Error (MPE) are commonly used. Similarly, significant interest has been shown in discriminative approaches to acoustic modelling. These models represent a *direct*, non-generative model of words/phonemes given the observations, as opposed to a joint distribution. The primary conceptual distinction between discriminatively trained generative acoustic models, and discriminative acoustic models therefore amounts to differences in the decoding process, with the differences in the generative and conditional approaches having an impact.

### 2.4 Language Modelling

The prior distribution $P(W)$ is represented by a Language Model (LM). This distribution is based purely on the word sequence, and is therefore typically estimated by performing simple counts of word frequency on the training corpus. LMs which include some word context/history ($n-1$ words in length) in the estimate, are referred to as *n-gram* LMs. These models represent the prior probability of a particular word at index $i$ in a sequence as $P(w_i|w_{i-1},\ldots,w_{i-n})$. Some care must however be taken when higher order n-gram models are estimated for a given vocabulary, and training data set size. Increased context lengths yield
less reliable n-gram statistics, as the number of representative examples for the
more infrequent contexts tails off drastically. Therefore, statistical smoothing
techniques and Back-Off models which default to lower order n-grams for more
infrequent contexts, are typically used.

2.5 Search

Finding the best word sequence for a given sequence of observation vectors was
formulated in terms of the MAP criterion in equation (2.3). Using this equation
directly represents a search problem. A brute force solution would be to firstly
enumerate all possible word sequence hypotheses. Thereafter, a network of HMMs
may be created for each such hypothesis by concatenating the word-level HMMs
together. The expression in the maximisation of equation (2.3) should then be
evaluated, which involves summing over all possible state sequences in the HMM
network to calculate the acoustic model term. The hypothesis under which this
likelihood is maximised represents the MAP solution.

In all but the simplest of ASR tasks the afore-mentioned approach is however
computationally infeasible, and in LVCSR more tractable solutions to the search
problem must be sought. A simplifying assumption that may be made is the
following:

\[ p(X|W) = \sum_{q} p(X, q|W) \approx \max_{q} p(X, q|W), \] (2.7)

where the state sequence \( q \) is the state sequence through all HMMs in the search
network. This is known as the Viterbi Approximation. The assumption here
is that the likelihood of the best path through the HMM network \( \hat{q} \) dominates
the sum in equation (2.7), and is therefore an adequate approximation of the
sum term. The search problem is now greatly simplified, as only the best state
sequence \( \hat{q} \) need be recovered from the HMM network.

The search space may be further constrained in LVCSR tasks by implementing
hypothesis pruning (also known as beam search). The result being that at every
time instant, the decoder essentially disregards state sequences which fall below
a certain \textit{likelihood threshold} or \textit{beam width}.

Owing to independence assumptions made by the HMMs in their representation of the output probability densities, the acoustic model likelihoods are typically underestimated. This bias has a significant impact on the MAP decision rule, due to the multiplicative operation performed on the probability distributions. These adverse effects are compensated for by the inclusion of an exponential language model scaling factor $\gamma$, and a word insertion penalty term, $\rho$. This yields the following improved MAP decoder function:

$$f(W) = p(X, q|W) P(W) \gamma \rho \|W\|. \quad (2.9)$$

\subsection*{2.6 Lattices}

A lattice is a graphical structure which is used to represent the pruned hypothesis space which results after running a recognition pass over the acoustic data. The lattice normally includes the hypothesised label (word and/or phone), the corresponding start and end times of the individual linguistic units, as well as the acoustic and language model likelihoods. Structurally, a lattice is essentially a directed, acyclic graph consisting of nodes and edges. Each \textit{edge} is labelled with the hypothesised linguistic unit, the start and end times of that unit, and the associated acoustic and language model likelihood scores. Each \textit{node} in the network corresponds to a point in time within the utterance. The identity of the start and end node for each edge is therefore used to impose the required structure. Figure 2.2 is an example of a word lattice for a short utterance.

This representation is very useful as many post-processing techniques can save computation time by having the alignment information for the more likely hypotheses in the search space readily available. Additionally, general graph search techniques may be applied without requiring knowledge of the probabilistic models used by the recogniser.
Figure 2.2: An Example Word Lattice

2.7 Evaluation

The most commonly used metric in evaluating the performance of LVCSR systems, is the word error rate (WER). This measure is based on the number of words which differ between the hypothesised transcriptions generated by the ASR system, and the reference transcriptions. However, these transcriptions do not necessarily have the same length. The comparison may therefore not be made by simply comparing the word identities in the two sequences at each index in the reference transcription. The sequences must therefore firstly be aligned using a dynamic programming (DP) procedure. This procedure is carried out so as to minimise the Levenshtein edit distance between the two sequences, which is defined as the weighted sum of occurrences of the following error types:

- **Substitution errors**, which represent the situation where a reference word is aligned with a *different* hypothesised word.

- **Deletion errors**, which represent the situation where a reference word is not aligned with *any* word in the hypothesis.

- **Insertion errors**, which represent the situation where an *additional word* is present in the hypothesis that cannot be aligned with a word in the reference.
The WER may then be calculated as the resultant number of errors divided by the number of reference words.

The WER metric discussed here only considers the best transcription hypothesised by an underlying ASR system (also referred to as the 1-Best transcription). Other metrics such as the oracle error rate, which is the lowest WER possible over a list of N competing transcriptions (referred to as the N-Best transcriptions), are also often used in evaluating LVCSR systems.
Chapter 3
Confidence Measures for Speech Recognition

A measure of confidence in a speech recogniser’s transcription is an invaluable source of information for many systems which incorporate an ASR engine. Out-Of-Vocabulary (OOV) detection and keyword spotting techniques can be based almost entirely on a measure of confidence. These applications therefore provided the initial motivation for the development of CE techniques. More recently, CE has garnered interest as a tool for use in dialogue systems, large scale transcription systems, system combination, and unsupervised adaptation of acoustic models.

The problem of estimating accurate measures of confidence for LVCSR systems is however a challenging one, and has by no means been solved. Improvements in the quality of confidence measures would undoubtedly have a significant impact in ASR systems, as the efficacy of most theorised applications of confidence scores has been hampered by the lack of truly accurate scores. In the sections which follow, a thorough review of the body of research in the field of CE will be presented.

CE approaches may be considered as falling into one of three broad classes [16], each of which will be discussed separately before being compared at the end of the section. As most of the existing literature is concerned with the estimation of word-level confidence measures, this will continue to be the form generally assumed in the discussions which follow.
3.1 Classification Approach

CE methods which fall into this category are concerned with the implementation of a post-classifier. The purpose of this classifier is essentially to estimate the likelihood that a word in the transcription was indeed transcribed correctly. The principle of the approach is that a hypothesised word may be classified as being correct/incorrect based on some single feature or set of features. These features embody information pertaining to the recognition process at the time the word was hypothesised.

3.1.1 Predictor Features

The set of features used in classifying word hypotheses are commonly known as predictor features. Many such features have been proposed in the literature, the majority of which are derived from information which is represented in recognition lattices. Theoretically, a predictor feature is any measure for which the distribution over correctly hypothesised words is significantly different from the distribution over incorrectly recognised words. Some of the more common classes of predictor features, along with examples from each class are described below.

- Features based on the lattice structure: Hypothesis Density [19] is a measure based on the assumption that the number of alternative arcs spanning the time segment for a word in the most likely transcription is indicative of the recogniser’s uncertainty in the hypothesis. This is an intuitive supposition as the recogniser normally prunes word hypotheses from the lattice in situations where the most likely word is significantly more likely than competing words. Word Trellis Stability [31] is a measure based on the premise that a word is more likely to be correct if it is present in a number of competing word hypotheses spanning a similar time interval.

- Acoustic-based features. The Acoustic Likelihood Scores normalised per frame or by the number of phones [27], is a natural representation of the match between the acoustic signal and the hypothesised word. Acoustic Stability [48] is a measure based on the number of times a given word
occurs in the same (aligned) position in $K$ different outputs from the recogniser. Where each of the $K$ outputs is generated using different values of the Grammar Scaling Factor (GSF) $\gamma$. This has the effect of weakening the coupling between the language model and acoustic model. Hypothesised words which are often aligned with the same position under different values for $\gamma$ are more likely to match the acoustics.

- **LM features.** A LM feature of particular interest described in [41] is the **back-off behaviour** (assuming a back-off LM is used by the recogniser). The premise here is that the language model is less confident in a particular word hypothesis when it has had to back-off to a lower order n-gram likelihood.

- **Duration-related features.** Measures based on the number of phones comprising the hypothesised word, and the duration of the individual phones, may be constructed from phone marked recognition lattices. Such measures were also suggested in [41]. The premise being that shorter phone sequences and words typically correspond to regions of poor performance by the acoustic model.

- **Word-level utterance features.** Measures such as word position, the length of the utterance, and the lexical identity of the hypothesised word are straightforward to compute. A classifier which employs such features will be similar in many ways to the Language Model used by the recogniser.

The majority of these predictor features are however not ideal in the sense that there is typically a significant amount of overlap between the resultant distributions for correctly recognised words and incorrectly recognised words. It has therefore widely been posited that the best approach is to combine predictor features in some way, so as to boost their individual discriminative power. Much of the literature which assumes this approach to CE is therefore primarily concerned with investigating different combinations of features, and classification frameworks under which to combine the individual features. A good representative work which explored the definition, combination and subsequent evaluation of a number of predictor features for CE is [3]. Statistical classifiers which have
been experimented with for predictor feature combination are Decision Trees, Maximum Entropy (MaxEnt) models, Generalised Additive Models (GAMs) and Support Vector Machines (SVMs) to name a few.

3.1.2 Shortcomings

Combining sets of predictor features has been shown to improve performance over single-score confidence measures in some cases. However, such approaches may only really be successful in improving performance when the individual predictor features are statistically independent. A study in [19] showed that most predictor features are in fact highly correlated. The result is that attempts at combining features do not yield significant gains over the best single predictor feature without considerable design effort. The design of a statistical classifier which can successfully combine multiple, arbitrary, potentially highly correlated predictor features to estimate a composite confidence measure is therefore an area of interest.

3.2 Posterior Probability Approach

The task modern ASR systems are faced with is to successfully recognise speech through the application of statistical principles. If speech recognisers were perfect, there would indeed be no need for confidence measures. The scale of the problem and the assumptions that must be made in typical ASR systems are the primary reasons recognisers produce errorful transcriptions. However, due to the statistical nature of speech recognition systems, it can safely be assumed that the recogniser’s own belief in the hypotheses it is assigning is a good indication of the true accuracy of the resulting best transcription. This theoretical viewpoint forms the basis of the posterior-based approaches to confidence estimation described in this section.

The principles of statistical ASR systems were discussed in Section 2.2. The MAP decision rule was formulated as choosing the hypothesised word sequence with the maximal posterior probability. This posterior probability may indeed be interpreted as a confidence score. The problem was however reformulated with
Bayes Rule to suit a generative HMM-based framework, and the denominator term was disregarded as it played no significant role in the decision rule (equation (2.3)). This is a common practice in HMM-based ASR systems, due primarily to the fact that explicitly modelling the distribution $p(X)$ would entail summing over all possible hypotheses. This is an intractable task for large vocabulary systems. The result is however that the posterior distribution is not actually modelled at all. Posterior probabilities must therefore be estimated by either making certain assumptions about the form of the distribution $p(X)$, or otherwise by employing approximate methods to estimate the distribution explicitly.

In the first class of solutions, so-called filler-based methods approach the problem by using a set of simpler, highly constrained filler models to represent the required distribution. Some example techniques are the use of all-phone recognition models [46], catch-all models [18], and making an approximation based on the highest word score assigned by the recogniser [5]. These techniques do however make fairly substantial assumptions, and little gain in performance has therefore been reported through their application.

The second broad approach, which aims to employ approximate methods to estimate the true distribution $p(X)$ has however proved more successful. Recognition lattices are compact representations of the most significant competing hypotheses generated by a recognition pass. As such, the hypotheses in a lattice would contribute the majority of mass to the afore-mentioned intractable marginalisation over all hypothesis/word sequences to estimate $p(X)$ directly. Computing posteriors over the lattice is therefore a fairly good approximate method. This does however still represent a computationally intensive task. In [30] and [42] the problem was further simplified by restricting the summation to the N-best hypotheses in the lattice. However, with more elegant algorithms and readily available compute resources - more general approaches which take into account a greater proportion of the hypothesis space from the recognition lattice are feasible. These Lattice-based posterior approaches will be described in more detail in the following sections.
3.2.1 Time-based Posteriors

A technique for estimating word-level posteriors on a complete word lattice was proposed in [10]. The first step in the proposed algorithm is to compute the posterior probability of each arc in the lattice. The likelihoods assigned by the Language Model (LM) and Acoustic Model (AM) are stored in the lattices and used to calculate these posteriors. Defining a path \( q \) through the lattice which includes the word \( \omega \) in word sequence \( W \), given the observation sequence \( X \), yields the joint lattice path probability:

\[
p(q, X) = p(X|q) \gamma P(W),
\]

(3.1)

where it should be noted that the scaling factor \( \gamma \) is used to scale down the acoustic model probabilities rather than scale up the language model scores, as is usually the case. This is necessary to ensure that the final joint distribution is not dominated by the highest-scoring path. The arc posterior probability can then be estimated by summing over all lattice paths which pass through the arc (the set \( Q_a \)):

\[
p(a|X) = \frac{\sum_{Q_a} p(q, X)}{p(X)}.
\]

(3.2)

The summation over lattice paths may be efficiently computed using the forward-backward algorithm. The lattice contains multiple arcs which have the same word identity, but correspond to slightly different temporal segmentations and n-gram contexts. In the case of estimating word-level posteriors, the segmentation and context are however not of interest. The next stage of the technique therefore involves the aggregation of arcs which correspond to the same word in the utterance. The process of determining which arcs should be considered equivalent is however not straightforward. The problem was originally addressed in the context of calculating log-likelihood ratios from N-best lists [40]. Potential clustering solutions were presented in [42], in which a word posterior distribution is defined over a specific time-frame corresponding to the same word. The posteriors are then computed based on the sum over the set of arcs which are considered part of the frame. The first method proposed in this work was \( C_{sec} \), in which the sum
is taken over all arcs with the same word identity that overlap with the current arc at all. $C_{med}$ restricts the sum to those arcs which have the same word identity but overlap with the median time-frame of the current arc. Finally, $C_{max}$ is essentially the same as $C_{sec}$, with the exception that instead of accumulating the posteriors, only the highest posterior of the overlapping arcs is selected as the posterior score. This work showed that the $C_{max}$ method yielded the best posteriors for use as confidence scores.

### 3.2.2 Confusion Network Clustering

Confusion networks [24] are a highly compact representation of the most likely hypotheses in a lattice. All paths through a confusion network are constrained to also pass through all nodes, resulting in a simple linear graphical representation of these hypotheses. The arcs in the confusion network correspond to words ($\epsilon$ arcs are added for null/missing words in the hypotheses). The nodes essentially impose a segmentation of the utterance into confusion sets. Figure 3.1 is an example of a typical confusion network. An algorithm for generating such confusion networks by clustering words into confusion sets is described in [24] and is summarised below.

![Figure 3.1: An Example Confusion Network](image)

1. The posterior probabilities for each arc in the lattice are calculated, using equation (3.2).

2. The posterior probabilities for each word in the lattice with a given start
time \((t_s)\) and end time \((t_e)\) are calculated using:

\[
P(\omega|t_{\text{start}}, t_{\text{end}}, X).
\] (3.3)

At this point, each word hypothesis \(\omega_i\) in the sequence \(\omega_0, \ldots, \omega_L\) of length \(L\) represents a singleton cluster, \(C_i\).

3. **Intra-word clustering** is performed. Clusters corresponding to the same word which overlap in time are merged. Poor clustering decisions at this point may have significantly detrimental effects on clustering decisions later in the process. To guard against this, candidate cluster pairs are processed in order of a posterior-weighted rank based on the degree of overlap between them.

4. **Inter-word clustering** is performed. Clusters corresponding to different words that are considered as belonging to the same confusion set are merged in this step. Clusters are considered for merging based on the time overlap between clusters, and a phonetic similarity score weighted by the posterior score. Care is taken to constrain the process so as to ensure the partial ordering (or precedence) of words imposed by the original lattice is not violated. This process continues until a total order of the clusters is achieved yielding the desired linear graph.

5. Finally, the \(\epsilon\) edges corresponding to null-words are added to the graph. The posterior probability assigned to this edge is defined as the remaining probability mass such that the sum of all posteriors in the confusion set sum to 1.

Confidence measures based on a piecewise-linear mapping of the posteriors obtained using this approach to posterior approximation are the current state-of-the-art. These posteriors were evaluated as confidence measures and used for lattice rescoring to minimise Word Error Rate (WER) on a Conversational Telephone Speech (CTS) task in [10] and [9].
3.2.3 Shortcomings

Lattice-based posteriors typically overestimate the true posterior distribution, as they are computed on a subset of the hypothesis space. They are also susceptible to the independence assumptions made by the recogniser. The raw posteriors must therefore be mapped to confidence scores in some way (as is discussed in [9]). This entails the implementation of an additional post-processing step before the confidence scores can be attached to the transcription. Furthermore, the heuristic nature of the arc clustering and consensus clustering approaches based on overlap and similarity scores is also not completely ideal, and is a potential source of error in the posterior estimation process. As posterior-based approaches have however generally proven to yield better performance than most alternative approaches, research into addressing these shortcomings is warranted.

3.3 Utterance Verification Approach

Drawing on research in the field of Speaker Verification, CE may be formulated as a statistical hypothesis testing problem within the framework of Utterance Verification (UV). UV is a post-processing approach much like the classification approach, in which the aim is to score the transcriptions after being generated by the ASR system. Early work in support of this approach was in relation to keyword rejection/keyword spotting systems [29].

3.3.1 Likelihood Ratio Testing

A given speech segment $X$ may be recognised by an ASR system as a particular word $w$, which is represented by the HMM $\Lambda_w$. The task of deciding whether to accept or reject this result may be formulated as a hypothesis testing problem. The null hypothesis, $H_0$, and alternative hypothesis $H_1$, may be defined as follows:

\[
H_0 : X \text{ was correctly recognised and is associated with HMM } \Lambda_w
\]

\[
H_1 : X \text{ was incorrectly recognised and is not associated with HMM } \Lambda_a.
\]
where $\Lambda_a$ is an HMM which corresponds to all alternative word hypotheses that are essentially incorrect. The Neyman-Pearson lemma states that under mild assumptions, the most powerful test which rejects one hypothesis in favour of another is the Likelihood Ratio Test (LRT). Modelling $H_0$ and $H_1$ with $\Lambda_w$ and $\Lambda_a$ respectively, the evidence under each distribution may be evaluated in order to yield a Likelihood Ratio (LR) expression of the form:

$$LR(\mathbf{X}, \Lambda_w, \Lambda_a) = \frac{P(\mathbf{X}|\Lambda_w)}{P(\mathbf{X}|\Lambda_a)} \geq \tau.$$  \hspace{1cm} (3.5)

In UV, a thresholding decision (with parameter $\tau$), is applied to this ratio in order to evaluate which hypothesis should be accepted. In CE this test is not of any real interest, and the LR score may be used as a confidence score, as proposed in [23].

The primary challenge in the UV approach is in the estimation of the alternate model $\Lambda_a$. Such models are also known as background or filler models when defined over all hypotheses, or anti-models when defined on a per-hypothesis basis. Alternative models are normally chosen to have the same HMM structure as the “correct” HMM $\Lambda_w$. This is the approach taken in [29], [32] and [28]. One unifying conclusion which can be drawn from the afore-mentioned publications, is that the alternative models should be trained within some discriminative framework in order to yield significant CE performance gains. Minimum Classification Error (MCE) was used as the discriminative training criterion in [28] while the Minimum Verification Error (MVE) criterion was implemented in the work by [32] and [29].

A related approach which assumes a Bayesian viewpoint and does not rely on the assumptions required to satisfy the Neyman-Pearson lemma is presented in [17]. This approach involves the computation of the Bayes Factors:

$$BF(\mathbf{X}, \Lambda_0, \Lambda_1) = \frac{\hat{p}(\mathbf{X}|H_0)}{\hat{p}(\mathbf{X}|H_1)} = \frac{\int f(\mathbf{X}|\Lambda_0, H_0)p(\Lambda_0|H_0)d\Lambda_0}{\int f(\mathbf{X}|\Lambda_1, H_1)p(\Lambda_1|H_1)d\Lambda_1},$$  \hspace{1cm} (3.6)

where $k = 0, 1$ are indices for the hypotheses, $\Lambda_k$ is the model parameter under the hypothesis $H_k$ and $p(\Lambda_k|H_k)$ is its prior distribution. The likelihood function for $\Lambda_k$ under $H_k$ is $f(\mathbf{X}|\Lambda_k, H_k)$. The expression in (3.6) represents the evaluation of
the evidence in favour of $H_0$. In much the same way as in the case of the likelihood ratio, the BF value may therefore be compared against a decision threshold $\tau$ for UV tasks, or may alternatively be used directly as a confidence measure.

### 3.3.2 Shortcomings

A significant challenge in UV-based CE is that of estimating the alternate models, as has already been mentioned. Such models represent highly complex composite distributions, and there is no clearly defined way to define these *alternatives* to the null hypothesis models. A truly alternative model may for instance be trained on different data, or may be based on a completely different modelling technique. This has been an ever-present challenge discussed in the literature on UV-based CE techniques, and is yet to be resolved.

### 3.4 Evaluation of Confidence Measures

A particularly useful evaluation method for word-level confidence measures based on principles of information theory, is the Normalised Cross Entropy NCE\(^2\) score, proposed in [33]. The NCE score is a measure of the amount of information which is gained through considering the confidence scores under evaluation, over the baseline of setting all scores equal to the empirical probability of a word being correct $p_c$. This baseline score on a dataset of size $N$, with $n$ correctly hypothesised words is simply $\frac{n}{N}$. The evaluation of a CE system therefore requires that the hypothesised word sequence be marked with the estimated confidence score, before it is aligned with the reference transcription. Each hypothesised word in the sequence is assigned a tag corresponding to whether it is indeed correct ($c_i = 1$) or incorrect ($c_i = 0$). The NCE measure may then be defined as follows:

\[
NCE = \frac{H(C) - H(C|X)}{H(C)},
\]

where $H(C)$ corresponds to the entropy of the tag sequence, and $H(C|X)$ is the entropy of the confidence score sequence. An NCE score of 0 indicates that

\(^2\)This measure is the standard CE evaluation metric employed by the NIST SCLite scoring tool
the confidence scores under evaluation provide no additional information (and are equivalent to the baseline). Positive scores indicate the confidence measures embody some additional information, and therefore represent an improvement in CE performance.

A graphical method often used in the evaluation of confidence measures which has its foundation in signal detection theory is that of Receiver Operating Characteristic (ROC) curves. In the binary classification task of CE, a false positive may be defined as an event whereby a word is classified as correct when it was indeed incorrect, and the definition of a true positive follows. The classification may be represented as a threshold decision on the confidence score assigned to each word. In constructing an ROC curve, the false positive rate is plotted against the true positive rate, given some operating point specified by the value of the variable threshold. ROC plots therefore represent a visualisation of the trade-off between rejecting correct hypotheses and accepting incorrect hypotheses. On such plots, curves which are above the $y = x$ line represent systems which perform better than the baseline.

A related plot known as the Detection Error Trade-off (DET) curve is also widely used in the literature for comparing confidence measures. These curves plot the false acceptance rate against the false rejection rate. However, DET curves typically make us of a non-linear scale on the axes - making the plots simpler to interpret. DET curves below the $y = -x + 100$ diagonal indicate system performance better than the baseline. Naturally, curves which approach the origin are optimal (0% correct words removed and 0% incorrect words retained).

Another metric which is sometimes quoted is the Equal Error Rate EER. This is a single score which specifies the operating point at which the false reject rate equals the false accept rate. Graphically, this corresponds to the intersection of the ROC curve with the $y = x$ line. Both DET plots and NCE scores will be used in order to visualise and evaluate the performance of confidence measures in this work.
3.5 Comparison of Approaches

There are a number of key areas in which the three approaches to CE can be compared and contrasted. Each of these is discussed separately in the sub-sections which follow.

3.5.1 Information Sources

UV techniques represent a flexible framework under which a number of knowledge sources/models can be combined to yield a composite confidence estimation framework. This is essentially also true for classification approaches with multiple predictor feature streams. The exception however being that many single-classifier solutions inherently need to assume independence of the input features in order to do so. Lattice-based posterior approaches do not make provision for any additional knowledge sources, and are based solely on information which is already available in the ASR system. It could however be argued that such information is the most useful for classifying hypotheses as being correct/incorrect. The crux of the argument being that considerable design effort is expended in designing these systems in the first instance, making this information more relevant.

The principle of being able to incorporate additional knowledge sources in an elegant manner through UV or a classification approach is nevertheless an attractive one. However, at some point such approaches stray into the domain of System Combination as the models which represent the additional information sources become increasingly complex. CE is typically intended to be a task that should require less design effort than in designing a speech recogniser itself.

3.5.2 Complexity of the Approach

Perhaps the most elegant solution to CE in terms of complexity is that of lattice-based posteriors, as no external model need be trained. Furthermore, the information required to estimate the posteriors is readily available in the lattice. Post-classifiers introduce an additional level of complexity as the relevant model needs to be trained, and additional data sources for training need to be con-
structed. The UV framework necessitates that *more than one* new model be trained on *separate* data sets. The challenges in defining a set of adequate models for UV suggests that this approach will be more complex than a classification approach.

In terms of system parameter tuning, the algorithms and heuristics involved in carrying out lattice-based clustering methods may require *some* time investment in tuning the parameters. Selecting the best combination of predictor features in classification approaches may represent a significant amount of design effort - unless methods can be found to circumvent this feature optimisation step. In a complex UV framework with multiple potential knowledge sources, this aspect of the complexity will be roughly the same as that of classification approaches.

### 3.5.3 Flexibility in Score Granularity

Lattice-based posteriors are typically defined at the word level, but may potentially also be calculated at the phone level if the lattice is marked up with the necessary phone labels. Such approaches are however unable to define Confidence Measures (CMs) at a linguistic unit level for which the information is not readily available in the lattice. However, classification approaches represent a flexible framework through which CMs may potentially be defined at arbitrary levels of linguistic unit granularity (e.g. at a sub-utterance *phrase* level). UV approaches could theoretically also model scores at different levels. However, the estimation of alternative models for these scores would represent a significantly more complex design effort than that of a similarly defined classification approach.

### 3.5.4 Modelling Additional Information

This is potentially the most significant motivation for classification approaches. The classification framework may be used to not only discriminate based on information similar to that used in recognition - but may also explicitly account for longer range effects in the data. An example of such an effect is the tendency of recognition errors to occur in *runs*. It is expected that a sequential classifier would inherently be able to model such a phenomenon. Lattice-based approaches are not generally able to explicitly model such effects. UV approaches may theoretically
be able to model such effects. However, this does entail the estimation of multiple models - each of which may be at least as complex as the single classifier in the classification approach.

### 3.5.5 In-system Application of Confidence Scores

An interesting application of lattice-based posteriors is that of lattice rescoring and decoding [9]. Rescoring a recognition lattice with the word posteriors is a relatively straightforward process, and may be implemented by augmenting the standard MAP decision rule with the posterior scores. This is however not possible within current classification and UV approaches to CE, as they do not operate on lattices directly, and assign scores to *transcriptions* generated by the ASR system.

Another in-system application of confidence scores, is in that of unsupervised acoustic model adaptation [1]. The idea in this approach is to use confidence scores as a means through which the execution of the relevant algorithms may be guided, such that speech segments which are to be considered more “correct” are used for adaptation. As such, lattice-based methods are significantly more suitable to such applications, as the posterior scores estimated in this fashion are reliant on the (potentially adapted) acoustic model scores, and therefore form part of a feedback loop. Post-classifier methods are not expected to be as suitable, as the test and training data characteristics will diverge under the adaptation framework, with the classifier not being able to adapt to these changes.
Chapter 4

Classifying Sequential Data

Sequence classification is a common problem across a number of fields. It is essentially pattern classification applied to data which is inherently sequential in nature. The aim being that of estimating the most likely sequence of labels $y$, given the sequence of observations $x$:

$$y^*(x) = \arg \max_y P(y|x).$$ (4.1)

Many statistical models for sequence classification (graphical models in particular), have been proposed in the literature. In the sections which follow, the characteristics of well-known sequence models will be discussed, before Conditional Random Fields (CRFs) will be presented as a particularly interesting class of sequential model.

4.1 Models for Sequence Classification

The graphical modelling framework makes it possible to efficiently model multiple interdependent variables. This broad approach is therefore significantly more powerful than simpler classification models which predict assignments of single class variables in isolation. The particular dependency that the graphical models discussed in this section exploit, is that of some sequential structure in the variables.
4.1.1 Generative Models

A generative statistical model is one which is able to randomly generate observable data according to some probability distribution. Such models generally specify a joint distribution over observations $X$ and labels $Y$. A Naïve Bayes (NB) classifier is the simplest form of generative model, and is concerned with generating observations $X$ for a single/atomic label $Y$. These models are in fact based on the the naïve Bayes assumption that all dimensions of the observations are independent. A reasonable approach to classifying sequential data might be to concatenate multiple NB models together, one for each index in a sequence of length $n$. This would however impose further independence assumptions between each label in the label sequence, and would not be able to leverage any information of the sequential nature of the data.

Under the assumptions that the current label depends only on the previous label (known as the first-order Markov assumption), and that each observation $X_i$ is only dependent on the current class label $Y_i$, a generative model with a linear chain structure may be defined. Such models are referred to as Markov Chains, the graphical structure of which is shown in Figure 4.1(a). The arcs in this structure are referred to as edges in the graphical model nomenclature, and the circles are referred to as vertices. Shaded vertices represent unobserved variables and the directed edges signify conditional dependence relations between the variables. Markov chains factorise the joint distribution over the label and observation sequences $Y$ and $X$ as a product of class-conditional output probabilities (or emission probabilities) and state/label transition probabilities between...
adjacent states. For the remainder of this section it will be assumed that the sequential labelling task concerns an observation sequence $\mathbf{X} = \mathbf{X}_1, \ldots, \mathbf{X}_n$ of length $n$. The corresponding label sequence is augmented with $Y_0 = \text{[start]}$ and $Y_{n+1} = \text{[end]}$ labels, yielding a label sequence of length $n + 2$. The Markov chain model therefore defines the following joint distribution:

$$p(Y, \mathbf{X}) = \prod_{i=1}^{n+1} P(Y_i | Y_{i-1}) p(\mathbf{X}_i | Y_i).$$

(4.2)

The parameters for such models are estimated by maximising the joint likelihood the model assigns to some training data (as expressed in equation (4.2)), using Maximum Likelihood (ML) training algorithms.

The application of HMMs to acoustic modelling was described in Section 2.3, in which the generative nature of the models and typical output distributions were discussed. An HMM can be thought of as a specialised form of the Markov Chains discussed here. The difference being that a one-to-one mapping between observations and labels is not assumed. Stated differently, the state sequence which generates a given observation sequence is not known. This leads to a hidden state representation. These hidden states may be marginalised out in order to recover the distribution over output variables in the generative sense ($\mathbf{X}$).

### 4.1.2 Discriminative models

For many sequence classification tasks, the generative framework is unnecessarily complex as the observation sequence is given. The implementation of a generative model which estimates a joint distribution is not necessary when the problem is a conditional one. The additional complexity in inherently modelling the distribution over observation sequences, $p(\mathbf{X})$, can degrade the performance of generative models. This is typically due to the inaccuracy of the naïve Bayes assumption. Discriminative models for sequence classification which directly model the conditional probability distribution of the data, $P(Y | \mathbf{X})$, are therefore interesting alternatives. As it is no longer necessary to make any assumptions regarding $p(\mathbf{X})$, it becomes possible for the sequence $\mathbf{X}$ to incorporate features which are
highly dependent/correlated.

The Maximum Entropy Markov Model (MEMM) [25], is an example of such a discriminative model. MEMMs replace the emission and transition probabilities in traditional Markov chains with with a single transition probability conditioned on the current observation, $P(Y_i|Y_{i-1}, X_i)$. This represents the probability of transitioning to state $Y_i$ given the previous state $Y_{i-1}$ and the current observation $X_i$. The resulting model architecture is shown in Figure 4.1(b), in which the unshaded vertices represent observed variables. An exponential model is used to represent the distribution $P(Y_i|Y_{i-1}, X_i)$, which is factorised as a number of feature functions.

The Maximum Entropy (ME) framework forms the basis for estimating the required parameters of the probability distribution, as expressed through the afore-mentioned feature functions. The Maximum Entropy principle states that the model which is the most appropriate, is that which is consistent with certain inherent constraints in the training data, while still making as few assumptions as possible. The primal problem may be cast as follows:

$$P^*(Y|X) = \arg\max_{P(Y|X) \in P} H(Y|X),$$

(4.3)

where $H(Y|X)$ is the conditional entropy of the label sequence, $X$, and $P$ is the set of all possible distributions consistent with the data. Consider a set of $k$ feature functions $f_1, \ldots, f_k$ which are representative of certain aspects of the underlying training data. A Moment Constraint (MC) may then be applied in parameter estimation which states that the expected value of each feature function in the estimated distribution, $E(f_k)$, be the same as the empirical expected value on the training data set $\tilde{E}(f_k)$. This constraint is simply:

$$E(f_k) = \tilde{E}(f_k).$$

(4.4)

Finding the distribution $P^*(Y|X)$ in equation (4.3) can then be formulated as a constrained optimisation problem using the moment constraints, standard probability density function (PDF) constraints and the definition of the primal problem. Carrying out this derivation yields the following expression for the distribution
modelled by a MEMM:

\[
P_{Y_{i-1}}(Y_{i}|X_{i}) = \frac{1}{Z(X,Y_{i-1})} \exp \left( \sum_k \lambda_k f_k (X, Y_i) \right),
\]

where \( P(Y_i|Y_{i-1}, X_i) \) has been split into \(|Y|\) separately trained distributions of the form \( P_{Y_{i-1}}(Y_i|X_i) \). The normalisation term \( Z(X,Y_{i-1}) \) ensures the distribution sums to one across all transitions from one state to the next. The parameters that are to be estimated are the weights \( \lambda_k \) for each of the feature functions. MEMMs have been successfully applied to a number of tasks. Most notably so within the field of Natural Language Processing, for Information Extraction and Segmentation [25].

A related discriminative model for sequence classification will be discussed in more detail in the following section, as it is particularly relevant within the context of this report.

### 4.2 Linear-Chain CRFs

#### 4.2.1 Definition and Derivation

Conditional random fields (CRFs) were first proposed in [20], as a discriminative modelling framework for segmenting and labelling sequence data. Following the definition given in the afore-mentioned work, the label sequence \( Y \) may be represented by a Markov Random Field (MRF) using a graph \( G = (V, E) \). \( V \) being the set of vertices in the graph, and \( E \) the set of connecting edges. The structure of the graph should be such that the label sequence \( Y \) is indexed by the vertices in the graph, \( Y = (Y_v)_{v \in V}. \) Then \( (X, Y) \) is a CRF if, when conditioned on \( X \), the variables \( Y_v \) obey the Markov property with respect to the graph:

\[
P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w \in N(v)),
\]

where \( N(v) \) is the set of neighbouring vertices for vertex \( Y_v \).

The graphical structure of the model may be segmented into a number of sub-graphs called *cliques*. Each clique, \( Y|_c \), is defined as a subset of the set of vertices
$C \subset V$, such that there is an edge connecting every pair of vertices in $C$. As equation (4.6) holds by definition, then by the fundamental theorem of random fields ([15],[20]), the joint distribution over the label sequence $Y$ given $X$ may be factorised into a product of potential functions, $\psi_c$, acting on each clique. The potential functions essentially perform the task of specifying the relation between the variables which make up the context of the clique on which it acts.

Although the structure of the CRF graph may indeed be arbitrary, attention will be restricted to linear chain CRFs in this section (illustrated in Figure 4.1(c)). Assuming this structure, the cliques are simply made up of the vertices and edges in the graph. Based on the general definition provided in the above discussion, the expression for the distribution modelled by a linear chain CRF may be expressed as follows:

$$P(Y|X) \propto \exp \left( \sum_{e \in E,k} \lambda_k f_k(e,Y|_e,X) + \sum_{v \in V,k} \mu_k g_k(v,Y|_v,X) \right), \quad (4.7)$$

where $Y|_s$ corresponds to a particular clique, and is therefore the set of components of $Y$ associated with the vertices in subgraph $s$. Furthermore, the feature functions $f_k$ and $g_k$ correspond to the exponential term of the potential functions $\psi_s$. The formulation in equation (4.7), and the associated nomenclature, will be preferred to that of the general graph theory treatment of potential functions and cliques in the remainder of the discussion on CRFs.

### 4.2.2 Properties of Interest

Some of the properties of CRFs which make them a particularly interesting class of models are summarised below.

- CRFs are discriminative in nature and therefore do not expend effort in modelling the distribution of the observation sequences. The observation sequence being the model inputs in the discriminative sense, which are generally fixed at test time for many tasks.

- It is possible for the conditional probability of the label sequence to depend on highly correlated complex input features without needing to explicitly
account for these dependencies.

- These input features can be completely arbitrary and may even be based on attributes of the observation sequence defined at different levels of granularity.

- The label sequence is conditioned on the entire observation sequence. This makes it relatively straightforward to model long term dependencies in the observation sequence.

- The Maximum Entropy training criterion applied to the exponential form of the potential functions simplifies training and inference algorithms significantly.

Although MEMMs (discussed in Section 4.1.2) share many of the characteristics mentioned above, they suffer from one crucial downfall: the label bias problem. This problem is a result of the fact that MEMMs employ per-state exponential models to represent the conditional transition probabilities between states. The result being that an exit transition from some state need only compete against other transitions from that same state, and not against all other transitions in the model. This per-state normalisation enforces that all probability mass arriving at a state must be distributed amongst successor states. Observations can therefore only affect which destination state is the most likely, and not the proportion of the probability mass which should be transferred to the successor state. This compromises the efficacy of the model, as this phenomenon results in a distinct bias towards states with fewer outgoing transitions. CRFs do not however suffer from the label bias problem, owing to the fact that they employ a single exponential model to represent the probability of the label sequence given the entire observation sequence. This greatly enhances the theoretical efficacy of the model, as the weights of the arbitrary features relating to different states can be traded off against each other across the entire model.

In summary, CRF models are able to condition on the entire observation sequence in a principled, probabilistic manner through the definition of arbitrarily complex feature functions acting on the entire observation sequence in an unnormalised fashion. This makes them a good candidate for application to many
sequence modelling problems.

4.2.3 Feature Functions and Input Features

It is of vital importance at this stage to establish a clear distinction between input features and feature functions. Input features refer to the observation feature vectors which serve as inputs to a CRF model (i.e. the sequence $X$). Feature functions are the functions defined by the model which may act on arbitrary combinations of input features/observations and output labels.

Within the Maximum Entropy framework, the set of feature functions should be constructed so as to describe certain characteristics of the underlying distribution to be modelled. Adequately engineered feature functions are likely to improve the modelling capability of a CRF, by explicitly incorporating features to represent potentially discriminative characteristics of the data. An example of a feature function from a Part of Speech (POS) tagging task may for instance be one which associates the label corresponding to proper nouns with observed words which are capitalised.

Due to the proliferation and proven efficacy of HMMs in many sequence classification tasks - the “base set” of feature functions which are commonly constructed are analogous to HMM transition features and emission features. Using this set of feature functions also ensures direct comparisons can be made between CRFs and HMMs. The transition features are defined for each pair of states in the model $y'$ and $y$, and correspond to the vertex features in the original CRF formulation of equation (4.7). The emission features are defined for each state-observation pair $y$ and $x$, and correspond to edge features in equation (4.7). These features may be expressed as follows:

\[
t_{y',y} = \delta(Y_{i-1}, y')\delta(Y_{i}, y) = f_{k,k\in\text{trans}}(Y, X, i) \tag{4.8}
\]

\[
g_{y,x} = \delta(X_{i}, x)\delta(Y_{i}, y) = f_{k,k\in\text{obs}}(Y, X, i), \tag{4.9}
\]

where the Kronecker delta function $\delta$ is used to indicate a match between its arguments (a proposed state/observation and the actual state/observation). With
equations (4.8) and (4.9) the distribution modelled by the linear chain CRF may be defined as follows:

\[
P_\theta(Y|X) = \frac{1}{Z_\theta(X)} \exp \left( \sum_{k,k \in \{y',y\}} \lambda_k t_k(Y,X) + \sum_{j,j \in \{y,x\}} \mu_j g_j(Y,X) \right),
\]

where \( \theta \) is the parameter set of the CRF. One difference which is immediately evident between this distribution and that of (4.5) for MEMMs, is in the normalisation term \( Z_\theta(X) \). The normalisation term in (4.10) is solely dependent on the observation sequence and not also on the previous label. This corresponds to a significant decrease in computational complexity when evaluating this term. The feature functions in the exponential term have been separated into transition and emission components for illustration purposes. However, as shown on the right hand side of equations (4.8) and (4.9), these may be combined into a single feature vector \( f \) having \( K \) independent features (and \( K \) weight parameters \( \lambda_1 \ldots \lambda_K \)).

### 4.2.4 Parameter Estimation

The parameters of the model, \( \theta = (\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_j) \) are typically estimated from a training data set \( D = \{X^{(p)}, Y^{(p)}\} \). Analogous to MEMMs, parameter estimation for CRFs is carried out within the framework of Maximum Entropy. In the discussion which follows, it will be assumed that the parameters for \( K \) feature functions must be estimated from a training data set \( D \) consisting of \( N \) labelled training patterns, with each such sequence being of length \( T \). Using the generalised feature vector representation, the distribution for a given parametrisation \( \theta \) may be restated as:

\[
P_\theta(Y|X) = \frac{1}{Z_\theta(X)} \exp \left( \sum_k \lambda_k f_k(Y,X) \right).
\]

38
The model parameters are estimated so as to maximise the log likelihood $L$ that the model assigns to the training data set:

$$L(\theta) = \sum_{p=1}^{N} \log P(Y^{(p)}|X^{(p)}) = \sum_{p=1}^{N} \sum_{i=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(Y_{i-1}^{(p)}, X^{(p)}, i) - \sum_{p=1}^{N} \log Z_\theta(X^{(p)}).$$

(4.12)

It should be noted that in practical implementations, a regularisation term should be added to the log likelihood function in equation (4.12). The role of this regularisation term is to guard against over-fitting, by penalising larger weight parameters. Optimising the partial derivative of this likelihood with respect to each weight parameter yields the maximal conditional likelihood solution. This partial derivative may be expressed as:

$$\frac{\partial L(\theta)}{\partial \lambda_k} = \sum_{p=1}^{N} \sum_{i=1}^{T} f_k(Y_{i}^{(p)}, Y_{i-1}^{(p)}, X^{(p)}, i) - \sum_{p=1}^{N} \sum_{i=1}^{T} \sum_{y, y'} f_k(y, y', X^{(p)}, i) P(y, y'|X^{(p)}).$$

(4.13)

When equation (4.13) is equated to zero, the resultant expression is the moment constraint formulation $E(f_k) = \tilde{E}(f_k)$, which is a desirable result.

In solving the parameter estimation problem, the empirical expectation $\tilde{E}(f_k)$ is trivial to compute as it merely involves counting the number of feature activations for each feature evaluated on the training data set. The computation of the normalisation in $P(y, y'|X^{(p)})$ involves summing over all possible state sequences consistent with the observation sequence, which is computationally expensive. This term may however be efficiently computed through the application of the forward-backward algorithm.

Although the function to be optimised is convex in nature - no closed form solution exists. In [20] Generalised Iterative Scaling (GIS) and Improved Iterative Scaling (IIS) gradient descent-based techniques are proposed for estimating the CRF model parameters. An analysis of training algorithms for CRFs in [39] clearly found IIS and GIS to be suboptimal, exhibiting slow convergence rates. Furthermore, this work showed IIS/GIS to be inappropriate for problems in which the sequence length is not fixed. Second order constrained optimisation techniques such as Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-
BFGS) were however shown in this work to be particularly suitable in parameter estimation for CRFs.

4.3 General CRFs

In Section 4.2, a specific linear structure was assumed for the CRF model. The most significant result of imposing this structure, was in the consequent definition of the graph cliques and the corresponding potential functions. For instance, the transition and emission features expressed in equations (4.8) and (4.9) act only on the vertices and edges which are associated with the current observation, current label and previous label (due to the linear chain structure). Although this structure may be suitable for many applications, a number of useful generalisations and extensions to the CRF framework have been proposed in the literature. These models are generally formulated by imposing less restrictive constraints on the graphical structure, such that the potential functions (i.e. feature functions), may be defined over more arbitrary contexts. Examples of such models which are of particular relevance to the work proposed in this report will be discussed in the sections which follow.

4.3.1 Higher Order/Skip-chain CRFs

Linear-chain CRFs are based on a first order Markov assumption. It is however potentially useful in many tasks to allow the model to capture dependencies between distant items in the observations sequence. By increasing the order $n$ of the Markov assumption, these long range dependencies may be modelled. However, this will typically result in an unnecessarily complex model, with the number of parameters to estimate increases dramatically as the order $n$ of the assumption is increased.

Skip chain CRFs are introduced in [35] as an elegant technique for modelling such long-range dependencies. The principle of this approach is to augment linear-chain CRFs with features that relate selected distant input features. This implies adding so-called skip edges to the model which link input feature vectors that are deemed similar. Furthermore, the feature functions acting on these edges may
incorporate arbitrary contextual information from each of the endpoints of the link. This input-specific model structure is only possible due to the non-generative nature of the CRF model. In [35] these models were proposed for the task of text segmentation, where words having the same identity were linked with skip-chains. This may however be generalised from a word-matching pre-segmentation of the input sequence, to an arbitrary matching function.

Considering the example of text segmentation, for a given sentence $X$, the set of all pairs of positions in the sequence for which a skip edge is to be defined may be represented as $J = \{(u, v)\}$. Augmenting a Linear-chain CRF with features to act on these edges yields the following re-formulation of the CRF expression:

$$P_\theta(Y|X) = \frac{1}{Z_\theta(X)} \left\{ \exp \left( \sum_k \lambda_k f_k(Y, X) \right) \prod_{(u, v) \in J} \exp \left( \sum_j \mu_j s f_j(Y_u, Y_v, X, u, v) \right) \right\},$$

where the parameters $\mu_j$ correspond to the $j$ skip edge features $sf$, and $\lambda_k$ parameters correspond to the $k$ remaining features in the model. The forward-backward technique already discussed for parameter estimation in linear-Chain CRFs, may be applied to a Skip-Chain model. However, if significantly many long-range overlapping loops are created through adding skip-chain links, approximate inference algorithms may be necessary in order to make parameter learning tractable in such situations.

### 4.3.2 Hidden-state CRFs

The general task of a pattern classifier is to map some observation vector $x$ to a corresponding class label $y$. However, in many sequential classification problems, a variable number of observations in the observation sequence $X$ should be considered as being related to a single label $Y_i$ at some index $i$ in the label sequence of length $l$. This mapping is unobserved in the training data, but may be modelled through a hidden/latent variable representation. Hidden CRF (HCRF) models [26] are an extension of CRFs which incorporate such a hidden state-space representation. Figure 4.3.2 shows a graphical representation of this model. The hidden variables correspond to the sequence $s$ in the figure. The dotted edges illustrate that the hidden variables $s$ may also act on longer term dependencies.
between input features.

The HCRF model defines a joint conditional probability of the label sequence $Y$, and a hidden state sequence $s$, given the observation sequence $X$ as $P(Y, s|X)$. In most sequential classification problems, it is not necessary to recover the hidden state sequence corresponding to a given label sequence. The hidden state sequence may therefore be marginalised out by summing over all possible state sequences $S$. This yields the desired posterior probability of the label sequence $Y$ as represented by a HCRF model with parameters $\theta$:

$$P_\theta(Y|X) = \sum_s P_\theta(Y, s|X) \propto \sum_{s \in S} \exp \left( \sum_k \lambda_k f_k(Y, s, X) \right).$$

(4.15)

The primary differences between this equation and equation (4.11) being the sum over hidden state sequences, and the inclusion of the hidden state sequence as a parameter in the feature function definitions. The parameters $\theta$ may once again be estimated within a Maximum Likelihood (ML) framework using the well-known forward-backward algorithm, in conjunction with a constrained gradient-ascent optimisation technique. Conjugate gradient (CG) and LBFGS are typically used in the literature for this purpose, and have both been shown to yield good convergence characteristics.

The definition of more complex feature functions which incorporate the hidden state sequence requires that significantly more parameters be estimated for such a model. The sum term over all possible states also has an impact on the complexity of the model. Nevertheless, the HCRF framework represents a powerful modelling
technique, and has been shown to consistently outperform HMMs in tasks for which they have historically been considered state-of-the-art. A particularly good comparison was presented in [34], where ML-trained and discriminatively trained HMMs were compared with HCRFs for the phone recognition task. The results of this work showed HCRFs outperforming state-of-the-art discriminatively trained HMMs. Such evidence warrants the investigation of their application to more tasks for which HMMs would typically be used.

4.3.3 Hierarchical CRFs

Hierarchical data structures occur naturally in a number of domains, and have been considered extensively within Natural Language Processing (NLP) and Imaging research. One way in which such hierarchies may be represented in a probabilistic framework, is through initially modelling each level in the hierarchy separately. A cascaded model which represents the entire hierarchy may then be built by using the output from each lower level as input to the level directly above it. There are however two particularly detrimental side effects of this approach. Firstly, errors introduced by the lower level models are propagated up to higher levels. This is known as cascading error. The second issue with cascaded model representations, is due to the fact that each model acts in isolation at a particular level. The result being that these models do not have any knowledge of potentially useful information from other levels. These shortcomings are the motivation for modelling approaches which attempt to model a joint representation over all levels within a single model. Hierarchical Hidden Markov Models (HHMMs) were proposed in [11] as a model having such a structure. The hierarchical nature of HHMMs is achieved through defining a nested HMM structure, in which each state in the model may be an HMM itself (up to a pre-defined depth). Hierarchical Conditional Random Fields are an alternative proposed in [37]. A similar line of reasoning is followed to that in the case of HHMMs, but within the undirected, conditional modelling framework of CRFs.

Central to the formulation of Hierarchical CRFs in [37] is the definition of a nested Markov process to a particular depth $D$. Such a structure imposes a model topology in which there are $S^d$ states at each level $d$ in the hierarchy $\{1..D\}$. Each
such state \( s^d \in S^d \) is the parent of a set of child states at the next level \( d + 1 \), \( \text{child}(s^d) \subset S^{d+1} \). Conversely, each child node has a set of parents, which may be shared amongst child nodes such that \( \text{parent}(s^{d+1}) \subset S^d \). The sharing of parents is necessary in order to avoid an explosion in the number of states for higher values of \( D \).

The nested Markov process with sequence length \( T \) then follows a temporal evolution pattern, in which the lifespan of child states are defined by that of their parents. Consider a state \( s^d_{i,k} \) defined on the interval \( \{i, k\} \) which is created at a level \( d < D \). This process leads to the instantiation of a child state \( s^{d+1}_i \) (at each level up to \( D \)). This process continues until some time \( j < k \), at which point a transition is made to a new state \( s^{d+1}_{j+1} \). This process of instantiating new child nodes continues until \( j = k \), and control returns to the parent state. The parent states may then either transition to a new state at the same level themselves, or if they too terminate may return control to their parent states in level \( d - 1 \). A graphical representation of the resulting model (based on that in [37]), is shown in Figure 4.3.3.

![Figure 4.3: Hierarchical CRF Architecture](image-url)

The structure of the model may now be specified. For each time instance \( i \) in the sequence, a node exists for each level in the hierarchy which represents the state variable \( x^d_{i} \in S^d = \{1, 2, \ldots, |S^d|\} \). Each state variable is augmented with an ending indicator \( e^d_{i} \), which is a binary feature used to indicate whether or not
the state variable is terminal \((e^d_i = 1)\) or should persist \((e^d_i = 0)\), based on the constraints defined below.

- The top state must persist for the duration of the sequence \(- e^1_{i:T-1} = 0, e^1_{T} = 1\).
- When a state terminates, all child states must also terminate \(- e^d_i = 1 \rightarrow e^{d+1:D}_i = 1\).
- When a state persists, all parent states must also persist \(- e^d_i = 0 \rightarrow e^{1:d-1}_i = 0\).
- When a state transits, all parent states must not change.
- The bottom state must not persist \(- e^D_i = 1, \forall i \in [1,\ldots,T]\).
- All states must end at time \(T\) \(- e^{1:D}_T = 1\).

The model architecture is therefore fully expressed through value assignments of the ending indicators.

The resultant graph can then be segmented into subgraphs, each of which corresponds to a particular context and a set of state variables. These are known as a contextual cliques and may belong to the following classes:

- State Initialisation cliques. Given the context \(c = (e^d_{i-1} = 1)\), a contextual clique \(\sigma^\text{init,d}_i = (y^d_i, y^{d+1}_i, c)\) specifies the act of a child state \(y^{d+1}_i\) at time \(i\) by a parent state \(y^{d+1}_i\).
- State persistence cliques. Given the context \(c = (e^d_{i-1;j} = (1,0,\ldots,0,1))\), a contextual clique \(\sigma^\text{persist,d}_{i;j} = (y^d_{i:j}, c)\) specifies the temporal span \([i,j]\) of a state \(y^d_{i:j}\).
- State transition cliques. Given the context \(c = (e^{d-1}_i = 0, e^d_i = 1)\), a contextual clique \(\sigma^\text{transition,d}_i = (y^{d-1}_{i+1}, y^d_{i+1}, c)\) specifies a transition from \(y^d_i\) to \(y^d_{i+1}\) at time \(i\) under the same parent \(y^{d-1}_{i+1}\).
- State Ending cliques. Given the context \(c = (e^d_i = 1)\), a contextual clique \(\sigma^\text{end,d}_i = (y^d_i, y^{d+1}_i, c)\) specifies the ending of a state \(y^d_i\) at time \(i\) with the last child state \(y^{d+1}_i\).
The Hierarchical CRF conditions the state variables $y_{1:T}^D$ and $e_{1:T}^D$ on the sequence of observations $X$ of length $T$. The relationship between these variables is captured through the definition of appropriate feature functions, which act on the contextual cliques described previously. Using the exponential form of the potential function these may be defined as follows:

$$
\pi_{u,i}^{d,s,x} = \exp\left(\sigma_i^{init,d} , X\right) \text{ where } s = y_i^d, u = y_i^{d+1}
$$

$$
R_{i:j}^{d,s,x} = \exp\left(\sigma_i^{persist,d} , X\right) \text{ where } s = y_{i:j}^d
$$

$$
A_{u,v,i}^{d,s,x} = \exp\left(\sigma_i^{transit,d} , X\right) \text{ where } s = y_{i+1}^d, u = y_i^d \text{ and } v = y_{i+1}^d
$$

$$
E_{u,i}^{d,s,x} = \exp\left(\sigma_i^{end,d} , X\right) \text{ where } s = y_i^d, u = y_i^{d+1}
$$

(4.16)

Given some observation sequence $X$, the probability of the label sequence $Y$ may therefore be expressed as:

$$
P(Y|X) = \frac{1}{Z(X)} \left\{ \prod_{d=1}^{D} \sum_{i_k,i_{k+1} \in \tau^d} R_{i_k,i_{k+1}}^{d,s,x} \right\} \times \prod_{d=1}^{D-1} \left\{ \sum_{i_k \in \tau^{d+1}, i_k \notin \tau^d} A_{u,v,i_k}^{d+1,s,x} \right\} \times \left[ \sum_{i_k \in \tau^{d+1}} \pi_{u,i_k}^{d,s,x} \right] \times \left[ \sum_{i_k \in \tau^{d+1}} E_{u,i_k}^{d,s,x} \right] \right\}
$$

(4.17)

where $Z(X)$ is the normalisation or partition function, and $\tau^d$ represents an ordered set of all end-time indices at level $d$.

Parameter estimation in Hierarchical CRFs can be achieved using an algorithm based on the Asymmetric Inside-Outside (AIO) algorithm originally proposed in [2] for HHMMs. The necessary adaptation of AIO from the generative formulation to for application to Hierarchical CRFs is described in [37]. This algorithm is relatively efficient and exhibits cubic-time complexity.
Chapter 5

Confidence Estimation on Lattices

Calculating confidence scores based on the information readily available in a recognition lattice is a particularly interesting approach to CE. It has a strong theoretical basis in statistical principles. A challenge in improving existing techniques lies in minimising the effect of heuristics/assumptions commonly employed by such techniques. The definition of a segmentation over words, and how competing hypotheses in the lattice may be aligned with this segmentation, are particularly challenging aspects of this approach. In the sections which follow, proposals for future work which are in line with this broad approach to CE will be discussed.

5.1 Direct Computation on a Lattice

Confidence measures may be estimated by executing an appropriate post-processing algorithm directly on the lattice. The convenience of not necessitating the design of additional models is invaluable. Estimating confidence scores based on an *entire lattice* was shown to outperform similar confidence scores restricted to N-Best lists generated from the lattices in [42].

Both confusion network clustering and time-based posterior calculation (discussed in Section 3.2.2 and Section 3.2.1) respectively, are reliant on the definit-
tion of some measure of overlap between words/lattice arcs. There are essentially two ways that the dependency on such heuristic definitions may be addressed. Firstly, generalisations can be made which are less restrictive in the enforcement of a temporal segmentation. This would yield an enhanced representation in which a distribution over possible temporal segmentations is defined, as opposed to a fixed segmentation. Another approach is to minimise/remove the dependence on the temporal information, by employing a technique which does not rely on temporal information to the same extent. Techniques which assume these approaches are described in the sections which follow.

5.1.1 CE using Lattice-based MBR

The MBR criterion is an alternative to the standard MAP criterion for ASR. The MBR estimate attempts to account for word-level accuracy through the use of an edit distance calculation in selecting the best hypothesised word sequence, \( \hat{W} \), as follows:

\[
\hat{W} = \arg \min_W \sum_{W'} P(W'|X)L(W,W'),
\]

(5.1)

where \( L(W,W') \) is the Levenshtein edit distance between the word sequences \( W \) and \( W' \). However, this optimisation presents a significant computational challenge. Particularly in the evaluation of the edit distance - which is a quadratic-time operation, and must be carried out for each hypothesis pair. The application of MBR to lattices was first proposed in [14], and has since found widespread use. However, N-Best approximations for the \( \arg \min \) and \( \sum \) terms in equation (5.1) are typically made such that the approach is indeed tractable (when using the standard form of the edit distance calculation).

Recent work in [44] re-visited the issue of Minimum Bayes Risk (MBR) estimation on lattices. The major contribution of this work is that the proposed technique operates over the entire lattice. This is achieved through the implementation of the proposed algorithm which carries out a forward-backward pass over the lattice, whilst simultaneously performing a dynamic programming (DP)-based edit distance calculation. A further contribution of this work is a method whereby a reference segmentation (based on the 1-Best hypothesis), can be re-
fined during the execution of the algorithm. This is a particularly useful result, as an ever-improving “best” segmentation is available for making alignment decisions/computing edit distances. The need for heuristic measures of overlap is therefore effectively removed. The matrix of statistics $\gamma(k, a)$, which is generated as part of the reference optimisation technique is of particular interest. Each element in this matrix essentially corresponds to a measure of the likelihood with which a word in the lattice, $a$, may be aligned with a position in the reference $k$, according to the MBR criterion. In effect, the temporal alignment of words across the lattice to each position in the reference is relaxed, in favour of defining a distribution over all words at each reference segmentation index. These statistics may therefore prove highly useful as individual confidence measures, or as predictor features for post-classification models.

The core of the approach as described above, is the (DP)-based Levenshtein edit distance calculation. The use of this loss function/criterion in the MBR framework, leads to word-level optimisation over the lattice, and indeed in the selection of the 1-Best reference used to calculate the $\gamma$ posterior statistics. However, the MBR framework allows for the use of arbitrary loss functions/criteria. A particular form of interest may be one which estimates a loss function over segments in the word sequences which was introduced in [13]. This is a particularly interesting approach, as it may be relevant in the estimation of phrase-level confidence measures. Further investigation into the application for CE is therefore warranted.

### 5.2 Direct Classification on a Lattice with CRFs

An interesting alternative to the direct application of post-processing algorithms to the lattice is provided in this section. CRFs are a class of discriminative models for classifying sequential data which can employ arbitrary feature functions, and are not sensitive to highly correlated inputs (see Sections 4.2 and 4.3). They have also shown to generally yield admirable performance across many different tasks.

The information stored in a lattice is highly sequential in nature, with each lattice representing multiple hypothesised sequences for a given utterance. Acous-
tic and language model statistics which represent these hypotheses in the lattice, are typically highly correlated. These factors play to the specific strengths of CRF models. The application of CRFs to estimate confidence scores directly on a lattice therefore represents an interesting research direction. For the discussion which follows it should be noted that the range of *input features* considered is limited to those that are already available in a typical recognition lattice.

There are two further fundamental principles which motivate the application of CRFs to CE based on information from a lattice. Firstly, a CRF model can account for long range dependencies in the sequential structure of hypotheses. Essentially, local decisions can be weighed off against those at subsequent or previous instances in the sequence. In the approaches discussed in Section 5.1 this was captured to some extent through the forward-backward computation of arc probabilities, and was an important aspect of these techniques. Secondly, a CRF model may be able to infer some representation of the alignment of competing hypotheses in estimating the confidence scores, without the need for explicit clustering or heuristic matching techniques.

The general sequential classification task involves finding the most appropriate sequence of labels $Y$ given a sequence of observations $X$. In a binary classification approach to CE, $Y$ corresponds to a sequence of *correct* and *incorrect* labels. The overall task is however assumed to be the estimation of *confidence scores* defined on the interval $\{0, 1\}$. Such a word-level confidence score may then be generated within this probabilistic framework through the implementation of the forward-backward algorithm to compute word-level marginal probabilities, as was proposed in [6]. The task for this system therefore becomes that of assigning a confidence measure to each word in the 1-best hypothesis (represented by $X$), where this measure is calculated as the marginal probability of assigning the *correct* tag to that word. Some key issues that must be addressed in implementing this approach are discussed in the following sections.

### 5.2.1 Input Features from the Lattice

The proposed system will make use of essentially the same input information sources as other lattice-based approaches. However - using *all* the information in
the lattice is not feasible in this approach. If it were feasible, it is unlikely that much would be gained through doing so, due to the discriminative nature of the approach. As the task is to estimate a measure of confidence and not necessarily an actual posterior probability for the hypothesised words, data corresponding to an N-Best list of hypotheses should be sufficient. One possibility would be to use input features such as the acoustic and language model likelihoods for competing hypotheses. This however requires the alignment of the N-Best hypotheses with the 1-Best hypothesis, and issues such as the representation of null words will need to be addressed. It should be noted that although the suggestion here is to consider word-derived features corresponding to the N-Best hypotheses, this is not to say that multiple lattice-derived features such as arc density may not be considered in addition.

5.2.2 CRF-based Clustering

Building on the N-Best representation discussed above, a more elegant approach would be to remove the need for the explicit alignment phase by expressing uncertainty in the segmentation through latent variables. The segmentation referred to here is that of the input features into spans which align with a single label in the label sequence. Representing an alignment in this manner performs a similar function to the clustering approaches discussed in Section 5.1. It is likely that such a model would benefit from the inclusion of the start and end times of word hypotheses as input features. Such information would no doubt prove discriminative in representing the alignment of the hypotheses.

The approach taken here is a departure from the heuristic approach, to one founded on machine learning principles. Although the clustering approaches do perform well and are indeed state-of-the-art, it is hoped that this approach may ultimately yield improved confidence measures. Hidden CRFs (see Section 4.3.2), have become well established in the literature and have been shown to provide significant gains in applications such as phone recognition [34], amongst others. Hidden CRF models should prove to be a useful modelling approach in implementing this method.

The CRF and Hidden CRF approaches described here and in Section 5.2.1
represent an additional modelling effort over competing approaches. They are however powerful, flexible approaches to the task of estimating confidence measures from lattices, and therefore warrant further research.
Chapter 6

Confidence Estimation with Rich Features

The approaches to CE discussed in this section differ from those in chapter 5 in one major aspect: the extent of the knowledge sources considered. The information available in lattices is not disregarded in the approaches presented here. However, methods through which additional/alternate sources of information may be applied are investigated. The motivation for this is based on the premise that a statistical ASR system may well be absolutely confident in the hypotheses it generates - even when this output is in fact incorrect. This is primarily as a result of the underlying statistical models used by the ASR system being incorrect, due in part to the effects of incorrect modelling assumptions. Information generated by the (potentially) erroneous models should therefore not be considered in isolation, and confidence measures should be estimated in a manner which incorporates external independent or derived information. These additional information sources will collectively be referred to as rich features.

6.1 A CRF-based Approach

CRF models are proposed here as the framework under which to combine the afore-mentioned rich features. This choice is based on the following factors:

- The task of estimating confidence measures for hypotheses is a sequential
one, as the data is a sequence of input features $X$ corresponding to each hypothesised word. As described in Section 4.2, CRFs are sequential classification models.

- Features which comprise the input feature vector $X$ may potentially be correlated. The fact that CRFs are insensitive to such correlations in the input data is central to the approach. The fundamental difficulty in combining rich features, as reported in the literature, has been in dealing with such dependencies between the input features.

- Complex arbitrary interactions between the features can be expressed through the design of adequate feature functions. This makes it possible to explicitly account for certain assumed underlying characteristics of the data.

- The discriminative nature of the framework is well suited to confidence estimation and confidence tagging. This is due to the fact that the fundamental task concerns the assignment of labels to observations. There is no need to indirectly model the distribution of the data, $p(X)$, as is the case in a generative approach.

### 6.1.1 Input Features

Combining features derived from the lattice such as those described in Section 3.1.1 within the framework of a CRF model, should already yield a fairly effective confidence estimation system. However, the true power of this approach lies in the combination of alternative sources of information.

As research in the field of ASR progresses, acoustic and language modelling techniques become increasingly sophisticated, yielding improved models. These improved models may be applied within a CE framework to score transcriptions generated by an existing ASR system. As this system may not necessarily incorporate models of the same standard, this therefore represents an interesting approach to CE. It should be noted that the intention here is not to carry out system combination, in which the aim is to combine hypotheses from competing “full” systems. The intention is rather to make use of an alternate model, generating relatively simplistic output. One particularly interesting example of such
an approach is that of using an alternative (potentially more accurate) acoustic model to generate an alternate “phone-only” decoding for an utterance. Such a phone-only decoding which is not constrained by the language model or grammar, was shown to be a useful predictor feature in [3]. Although improved language models may also be used as an information resource, it is already common practice to re-score lattices using improved language models, which would essentially amount to the same thing.

The aim of the approach described here is to construct a framework in which it is possible to incorporate any additional information source, without fear of making independence assumptions between the information generated by each such source. A truly flexible system should be able to incorporate arbitrary combinations of features/variables having different base types. Typical base types may for instance be text, discrete values, and real valued continuous values.

6.1.2 Complex Feature Functions

In Section 4.2.3, the base set of HMM-like feature functions were presented and mention was made of the fact that arbitrary feature functions may be constructed. The process of constructing such arbitrary feature functions is know as feature engineering. Feature engineering essentially involves representing some underlying truth in the data through some function acting on the appropriate input features and output labels. These engineered functions will be referred to as complex feature functions in this report.

Under the maximum entropy framework of CRFs, an assertion is made that the expectation of each feature function on the empirical data be the same as that represented by the model. Therefore, better engineered features which represent complex relations in the data allow the model to embody some representation of these complexities. This results in a more accurate model of the underlying distribution. There is a great deal of information in the sequential nature and structure of syntax, grammar and semantics of language and speech. This is relevant for CE as local decisions on the quality of transcriptions may therefore be greatly influenced by information pertaining to other parts of the utterance. Feature engineering is a framework through which aspects of these structural
dependencies may be exploited, and therefore represents a potentially rich area for experimentation within the proposed approach.

6.1.3 Hierarchical Representation

Speech and language is a domain in which the data is generally well-structured. The usefulness of a sequential model such as a CRF for sequentially structured data has already been discussed. Another structural aspect of speech and language data is its hierarchical nature. This is evident when one considers that a sentence or utterance comprises a sequence of words, each of which has a syllabic pronunciation, which may be further decomposed into individual phonemes. Each level of abstraction in this hierarchy also imposes a strict sequential ordering on its sub-units.

This hierarchical structure may be exploited to yield more powerful CE models/techniques. Complex feature functions and rich features discussed in the previous sections represent a framework through which features defined at different levels of abstraction may be combined. However, within a standard linear-chain CRF framework, considerable design effort would be required to construct feature functions that define the abstraction relationships between such multi-level input features.

A more elegant approach to incorporating features corresponding to different levels of granularity, is to make use a hierarchically structured model. In such a model, each level in the hierarchy is able to use information from lower levels in representing data at its own level. The effort in feature engineering is therefore reduced, as the model itself will learn the relation between the input features across different levels of abstraction. Hierarchical CRFs [37] exhibit this behaviour, as discussed in Section 4.3.3. The ability to combine rich features at different levels of granularity, and have the model learn to leverage aspects of the imposed hierarchy is one worth investigating. It should however be noted that some pre-determined hierarchy must be assumed in applying this modelling technique.
6.1.4 Phrase-level Measures of Confidence

The discussion thus far has been concerned with the estimation of word-level confidence measures. Although word-level measures are generally useful, there are applications in which a measure defined at a phrase level may be more useful. The definition of a phrase considered here is not a strictly grammatical one, but rather a grouping of consecutive words which represents some logical segmentation of the utterance. Two particularly interesting application areas for such a measure are described below.

- In statistical spoken dialogue systems, confidence scores already play a central role in ensuring robust operation [36]. However, the mapping from the word-level to semantic concepts and dialogue acts is a challenge. Confidence measures attached to spans of words or phrases would simultaneously impart a measure of confidence, and a notion of a sequence over which words in the utterance are to be considered similar.

- In mixed-mode transcription systems, where human transcribers are used to improve poor automatic transcriptions, such measures are also useful. Information which makes it possible to identify segments of utterances which should be manually transcribed may reduce the volume of referrals to human transcribers. Furthermore, this has beneficial privacy implications, as human transcribers will not necessarily be required to listen to and transcribe the entire poorly transcribed utterance.

There is very little published work which addresses the problem of confidence estimation at a level of granularity other than at the word-level. This is largely due to the fact that ASR systems are evaluated based on their word-level performance. The definition of error regions discussed in Section 1.1 facilitates the definition of a confidence measure over a phrase comprising consecutive words.

In existing work which incorporates some measure of phrase-level confidence, these measures are obtained by combining word-level scores over given phrasal segmentations. One technique through which these phrase segmentations may be obtained is detailed in [21]. In this work, the transcribed utterance is pre-processed with a parser. The resulting parse sub-trees are subsequently considered
to be the phrasal spans over which word-level confidence measures are combined. Such an approach may represent a first step in estimating such confidence measures for the purposes of the work proposed in this report. It would however be more interesting to investigate the possibility of removing the need for indirection through word-level measures.

A more elegant approach to the problem is based on the hierarchical architecture described in Section 6.1.3 and 4.3.3. By defining the top level of the hierarchy as being the phrase-level confidence labels, a confidence measure at the desired level of granularity may be modelled. Additional prosodic features acting at lower levels in the hierarchy, and grammatical features at the word level may then be defined as a means through which the model may capture some notion of the phrase boundaries. As has already been mentioned, a pre-defined fixed hierarchy must however be assumed. This presents a significant challenge as it is unclear how to represent the segmentation of a phrase into lower levels of the hierarchy within such a structure. Semi-Markov hierarchical CRFs [38], are an extension of the hierarchical CRF framework which may be better suited for this task. The graph structure is essentially adapted on-the-fly, the result of which is that arbitrary segment durations may be modelled at various levels of the hierarchy. This allows an arbitrary segmentation of the input features into a label sequence of unconstrained length. Furthermore, algorithms for semi-supervised learning of such models are presented in [38]. This approach is therefore well-suited to the problem of phrase-based confidence scoring, and may warrant further interest.

An evaluation measure for phrase-level accuracy will be required if such a confidence measure is to be investigated. Furthermore, the resultant definition of a “phrase” is central to the generation of suitable training data. As there is no suitable measure which has formally been described in the literature, any research into phrase level measures will have to be preceded by the development of such a metric and definition of the “phrase”. A metric which is based on similar information-theoretic principles as the standard word-level confidence measure described in Section 3.4 (NCE), would be desirable.
Alternate models may be considered as information sources which generate rich features. UV is a formalism through which such information sources can be combined for the purpose of estimating confidence measures. The UV approach requires that the models $\Lambda_w$ corresponding to the null hypothesis, and $\Lambda_a$ corresponding to the alternative hypothesis be trained. The modelling approach typically used is the same as that of the underlying ASR system being scored (e.g. HMMs). However, a recurring theme in much of the work which has assumed this approach, is that the models should be trained *discriminatively* for the approach to be successful. Although proven discriminative training algorithms for models such as HMMs are available, HMMs are still fundamentally *generative models*. Fundamentally *discriminative models* such as CRFs are therefore potentially better suited for modelling the null and alternative hypotheses. UV which implements CRF models is therefore put forward as an interesting area of research.
Chapter 7

Implementation of a CRF Toolkit

A number of experiments proposed in this report for future work involve the implementation of a particular class of discriminative sequential statistical model, the CRF. Open source software implementations of CRFs are available. However, these tools have generally been developed with the intention of being applied to tasks within NLIP, such as POS tagging and text segmentation. The implication being that these tools have limited support for continuous valued inputs (as they are designed for discrete/textual input). Such input features are however expected to be used extensively in the experiments described in this work. This fact, coupled with the desire to be able to experiment with variations of the general CRF model framework, and implement complex feature engineering, prompted the development of a bespoke CRF toolkit.

This chapter serves as a brief overview of the cross-platform CRF toolkit which was developed in C# as part of this work. The toolkit was developed in a flexible manner so as to provide a rich framework within which to carry out experimentation. Particular strengths of the tool are in the support for continuous input features, multi-modal input feature vectors, and complex feature functions. It is however the intention that this toolkit be expanded and developed as necessary while carrying out the required experiments, with increasingly sophisticated model hierarchies and feature functions being supported. Algorithms for both the efficient training and testing of CRF models have been implemented, yielding a complete end-to-end framework. At a system level, the tool has been developed such that execution may be parallelised over multiple processes or mul-
tiple threads, making it possible to exploit execution on a compute stack or on multi-processor machines.

It should be mentioned that the implementation was tested extensively during development. The toolkit was used to train and evaluate POS tagging models, and performance consistent with existing approaches and standard CRF toolkits for POS tagging was achieved.

7.1 Data

One of the main arguments for the use of CRF models in the proposed experiments, are as a result of their favourable properties with respect to multiple overlapping input features. It was therefore imperative that the CRF toolkit be able to support complex datasets, for which adequate provision was made. In particular, complex input feature vectors may be constructed, which represent arbitrary combinations of variables having different base variable types (e.g. word-id strings with continuous numbers). In this sense, the input feature vectors may be multi-modal. As CRF models are essentially sequential in nature, it is assumed that the datasets consist of sequences of input features, and their corresponding label sequences. However, datasets may also be specified which are not sequential in nature, for which there is one observation vector per label. Models trained on such data will essentially be atomic Maximum Entropy classifiers. The format of a specific dataset in terms of the observation vector template and sequence delimiter characters, is assumed to be specified in a key file accompanying each dataset. When continuous input features are specified, an additional processing step is applied to such data so as to make it compatible with the approaches described in Section 7.2.2. This involves a transformation which maps the data to the interval \([1,2]\). A sigmoid function is applied to the data, before an offset mapping in order to achieve the desired transformation. It is however assumed that additional pre-processing and normalisation of the data such as standardisation has already been carried out. Standardisation in this sense refers to the mapping of a continuous distribution such that it has zero mean and unit standard deviation. This is a particularly important normalisation step in systems which use multiple continuous data sources having different
distributions.

7.2 Feature Representation

Another argument in favour of applying CRF models, is that of the ability to engineer feature functions which act on the input data in a potentially arbitrary manner. This is of course in addition to the “base” set of features which correspond to transitions between labels, and relate the input features to labels. However, in order to successfully implement any such feature functions requires a complementary internal representation of the input features/data. Depending on the base type of the input feature, the corresponding feature functions may be represented as is described in the following sub-sections.

7.2.1 Discrete Features

Discrete numerals and text are the input feature data types on which these feature functions operate. Discrete feature functions are based on the Kronecker delta function, which performs a binary equivalence check between its arguments. Transition feature functions between labels and “emission” feature functions which relate word identities to labels, are examples of feature functions which act on discrete features. These features have a sensible representation within the maximum entropy framework, through the definition of moment constraints. The majority of CRF feature functions therefore take on this form, as most of the features in the data are either inherently discrete, or may be approximated as discrete in some way.

7.2.2 Continuous Features

Continuous inputs are of particular interest in this work. This class of input feature does not however have a clearly defined representation within the maximum entropy framework. This is evident when one considers that the fundamental task of designing a system such that $E(f_k) = \hat{E}(f_k)$ (a moment constraint), is not well defined when the features act on continuous distributions. Intuitively, the moment constraint applied to continuous variables entails that only the mean
value (or moment), of the feature distributions are constrained to be the same. This is however by no means a strong enough constraint for system design, as was discussed in [47]. The techniques described below are proposed solutions to this problem.

### 7.2.2.1 Moment Constraints with Binning

The simplest approach to the representation of continuous features is to generate feature functions for continuous inputs which make it possible to apply the standard moment constraints in some way. A technique described in [47] is to represent the data with quantisation features. These features effectively reduce the problem to one in which a set of discrete features is constructed, each of which corresponds to a specific quantisation sub-interval (or bin) on the overall interval for the continuous value. A quantisation feature $f_i$ defined on the interval $[l_i, h_i]$, may then be converted into $K$ binary features as follows:

$$f_{ik} = \begin{cases} \frac{h_{ik} + l_{ik}}{2} & \text{if } f_i \in [l_{ik}, h_{ik}] \\ 0 & \text{otherwise} \end{cases}, \quad (7.1)$$

where $k$ is the bin index.

Quantised bin features of this form have been implemented as part of the toolkit. As was already mentioned, the data is pre-processed such that all data-points are on the interval $\{1, 2\}$. A uniform occupancy binning approach is the default technique applied in defining the span of each of the $K$ bins (as in [43]). The value of $K$ may be specified as an argument to the toolkit when processing such features. A mode is also provided whereby the default binning technique may be disabled in favour of one which simply assumes a uniform segmentation of the interval.

### 7.2.2.2 Distribution Constraints with Cubic Splines

Assuming that an infinite number of training samples are given, the number of bins $K$ may be increased to infinity. By noting that only a single value of $f_{ik}$ will
be non-zero, and will take the value $f_i$ the following expression holds:

$$\lim_{K \to \infty} \sum_k \lambda_{ik} f_{ik} = \lambda_i(f_i) f_i,$$

(7.2)

where the parameter $\lambda_i(f_i)$ is no longer a single value as in the expression of equation (4.11), and is a potentially nonlinear function of the continuous input feature value $f_i$. This fact makes the parameter estimation problem for such features challenging.

However, a useful technique which solves the problem by using cubic splines to approximate the continuous weight function is presented in [47]. With $K$ knot points in the spline, a parametrised cubic spline approximation $a_k$ may be derived for every knot $k$. As detailed in [47], the product $\lambda_i(f_i) f_i$ may then be approximated as a sum of the products of $K$ transformed features in the form $a_k(f_i) f_i$, weighted by $K$ single-valued weights $\lambda_{ik}$. Under this approximation, the parameter estimation problem becomes tractable. This is due to the fact that the term $f_{ik} = a_k(f_i) f_i$ is only dependent on the continuous value of the feature and the location of the knots, and is independent of the weights to be estimated. Such feature functions result in Maximum Entropy constraints for which the distributions are matched, and are therefore known as Distribution Constraints (DCs). Continuous feature functions defined in this manner have been shown to improve modelling accuracy over the binning technique detailed in Section 7.2.2.1 on a number of tasks [47].

This representation of continuous features has been implemented as part of the CRF toolkit. The number of knots in the spline may be specified as a parameter for the feature generation process. The manner in which the spline knot points are placed may also be specified. The options are to place knot points based on uniform occupancy, a uniform segmentation of the interval, or such that the approximation error is minimised over the training data. The latter placement technique involves an optimisation step, and may lead to overfitting but is nevertheless interesting for comparison with the standard techniques.
7.3 Parameter Estimation

The CRF toolkit may be executed in *training mode* to estimate the parameters of CRF models. As was mentioned in Section 4.2.4, the original work on CRFs [20] suggested iterative scaling techniques to estimate CRF model parameters. In the initial development stages of the toolkit, IIS was implemented for parameter estimation. However, the slow convergence rate and sensitivity to arbitrary sequence lengths reported in [39] were encountered. As was suggested in the afore-mentioned work, the L-BFGS constrained optimisation algorithm was subsequently implemented.

The implementation of L-BFGS distributed in the ALGLIB numerical library was used to carry out the optimisation of the CRF objective function. This is a port of the original FORTRAN implementation presented in [22] to C#. The form of the objective function used includes a euclidean norm regularisation term of $1/2\sigma^2$ to reduce the effects of overfitting. The $\sigma$ parameter for the regularisation term may be specified as an argument to the toolkit. Successive iterations of the optimisation are carried out until convergence of the training data set likelihood is reached. The $\epsilon$ value for this convergence criterion may be specified as a parameter. A cross validation data set may optionally be specified, in which case training will halt when the likelihood of the cross validation data set stops increasing. Model parameters are written to file when training is complete, but may optionally be written to file at every optimisation step. This makes it possible to recover a particular model training process, should it be terminated prematurely for some reason.

7.4 Scoring Transcriptions

The CRF toolkit may also be run in *test mode*, in which a specific CRF model is applied in order to classify the data in a *test dataset*. There are two further variants of the test mode, each of which is discussed below.
### 7.4.1 Word Sequence Tagging

One mode of operation for testing is that of *tagging* each word in a given speech utterance as being *correct* or *incorrect*. The testing problem in this case becomes that of finding the *most likely* sequence of labels the CRF model would assign to the word sequence. The Viterbi algorithm was implemented as part of the toolkit in order to carry out the task of finding the best hypothesised label sequence efficiently. If the supplied dataset includes tags corresponding to the reference labelling of the words, the classification error is calculated and is output along with the list of confidence label hypotheses.

### 7.4.2 Word-level Confidence Measures

A second mode of operation which is more relevant for the task of CE, is that of estimating the likelihood of assigning the *correct* label to each word in the utterance. As was discussed in [6], this may be computed as a marginal probability estimated using the forward-backward algorithm. A forward backward pass is therefore carried out on each utterance in this mode, with the marginal probability $P(\text{correct})$ being evaluated for each word in the sequence. These statistics are then written to an MLF$^4$ file corresponding to the test input, for evaluation purposes.

---

$^4$A Master Label File (MLF), is the standard output format for transcriptions in HTK format.
Chapter 8

Experiments

8.1 Data Sets

The datasets used in these experiments were compiled using transcribed output (in the form of recognition lattices) from the CU-HTK 10xRT P1-P2 evaluation system, described in [8]. This system was evaluated on datasets derived primarily\(^5\) from the Fisher CTS Corpus [4], the details of which are summarised in Table 8.1. The WER quoted is that calculated over the 1-Best transcriptions generated by the system.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Hours</th>
<th>#Utterances</th>
<th>#Words</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev04</td>
<td>3</td>
<td>9044</td>
<td>37834</td>
<td>16.3</td>
</tr>
<tr>
<td>Eval04</td>
<td>3</td>
<td>7800</td>
<td>36781</td>
<td>19.0</td>
</tr>
<tr>
<td>Eval03</td>
<td>6</td>
<td>16681</td>
<td>76157</td>
<td>19.7</td>
</tr>
</tbody>
</table>

\(^5\)The Eval03 dataset includes utterances from the Switchboard corpus.
8.2 Lattice-Based Posterior Experiments

8.2.1 Posterior Estimation

For each of the datasets detailed in Table 8.1, a corpus was generated containing posteriors of each type experimented with in this work. These posteriors were obtained using the following implementations of the relevant estimation techniques on the lattices:

- The confusion network posteriors were estimated using the HTK\cite{45} based implementation used in \cite{10}.

- The time dependent word posteriors were estimated using the HTK based implementation⁶ used in \cite{9}.

- The Lattice-based MBR posterior statistics were estimated using the HTK based implementation⁷ used in \cite{44}. Modifications were however necessary in order to accumulate the desired statistics (γ).

Each corpus was subsequently scored using SCLite. The resulting SGML scored output file was parsed in order to construct an aligned parallel corpus of word-level posteriors and the corresponding correct/incorrect score tags. Experiments based on these posteriors are described in the sections which follow.

8.2.2 Baseline Definition

Two baselines were considered for each of the lattice-based posterior approaches. The first baseline corresponds to the evaluation of the “raw” posterior scores. These posteriors typically overestimate the true measure of confidence. Therefore, a secondary baseline was constructed, in which these posteriors are mapped to a confidence measure using a piecewise-constant approximation. This baseline corresponds to a relatively straightforward approach to generating confidence scores from word-level posterior scores, using a direct mapping function which

\footnote{The geometric mean metric \(C_{med}\) described in Section 3.2.1 was used as an arc overlap measure.}

\footnote{This implementation was kindly supplied by one of the authors of the cited paper, D. Povey.}
does not require significant design effort. A binning approach is used to estimate the baseline scores as follows:

- Given the parallel training dataset, a pre-processing step is performed in order to standardise and map the posterior scores to match the internal representation used by the CRF-based approach (see Section 7.1).
- The mapped score interval $[1, 2]$ is segmented into $i$ binning intervals.
- The statistics on the training data are accumulated over each such interval $i$ as follows:

$$
\hat{P}_i(\text{correct}) = \frac{\sum_{\omega \in \beta_i} \delta(y(\omega), \text{correct})}{\sum_{\omega \in \beta_i} \delta(y(\omega), \text{correct}) + \sum_{\omega \in \beta_i} \delta(y(\omega), \text{incorrect})},
$$

(8.1)

where $\beta_i$ is the set of training data words which have posterior scores that lie on the interval specified by bin $i$, and $y(\omega)$ is a function which returns the correct/incorrect tag for the hypothesised word $\omega$.

The statistics accumulated represent the desired mapping of the posterior scores into confidence scores. In test situations, the data to be scored is pre-processed in a manner consistent with that of the training data. A subsequent post-processing step then applies the afore-mentioned mapping estimated on the training data to the test data. In this manner, posterior scores on each interval $i$ are assigned the corresponding empirical posterior probability $\hat{P}_i(\text{correct})$.

### 8.2.3 Baseline Evaluation

The NCE scores for the baseline evaluation experiments are given in Table 8.2. The abbreviations CNP, TBP and MBR-$\gamma$ correspond to the confusion network, time based, and MBR posterior scores respectively. For ease of comparison, a uniform interval segmentation was used for all approaches. The decision to use a quantisation parameter of 8 for these experiments is based on [9], in which it was determined that this number of quantisation levels was adequate for mapping the posterior scores$^8$.

---

$^8$A decision tree was trained on the data, resulting in 8 leaf nodes which correspond to the decision boundaries which minimise the classification error.
Table 8.2: NCE Scores for Posterior Experiments on Eval04

<table>
<thead>
<tr>
<th></th>
<th>CNP</th>
<th>TBP</th>
<th>MBR-γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>18.6</td>
<td>19.0</td>
<td>18.6</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.295</td>
<td>0.325</td>
<td>0.293</td>
</tr>
<tr>
<td>Mapped (8 Bins)</td>
<td>0.317</td>
<td>0.347</td>
<td>0.328</td>
</tr>
<tr>
<td>CRF (8 Bins)</td>
<td>0.317</td>
<td>0.347</td>
<td>0.328</td>
</tr>
</tbody>
</table>

8.2.4 CRF Parameter Experiments

The effect of changing the CRF parameters with respect to the internal representation of continuous features was investigated. The confusion network posteriors were used as input features for these experiments. The aim of these experiments is to gain an understanding of the effect of these parameters, and arrive at a suitable configuration for subsequent experimentation. The parameters experimented with were the following:

- The technique used as an internal representation of the continuous input features. Namely, binning with moment constraints (denoted by B in the table), or cubic splines with distribution constraints (denoted by S in the table).

- The quantisation parameter of the binning/cubic spline technique. This corresponds to the number of bins used, or the number of knot points, for the two techniques respectively (the columns in the table).

- The algorithm used in determining the quantisation boundaries/points. The options being a uniform interval (denoted by E in the table,) or uniform occupancy segmentation (denoted by A in the table).

NCE scores for these experiments are given in Table 8.3, and a DET plot for some representative systems is shown in Figure 8.1. The regularisation parameter σ (see Section 7.3) was also experimented with. However, these results are not explicitly reported here, as changes in this parameter did not have a significant impact on the performance of the resulting system. A similar observation was made in [39].
### Table 8.3: NCE Scores for CRF Parameter Experiments on Eval04

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>8</th>
<th>15</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF+B+E</td>
<td>0.302</td>
<td>0.317</td>
<td>0.330</td>
<td>0.335</td>
<td>0.338</td>
<td>0.336</td>
</tr>
<tr>
<td>CRF+B+A</td>
<td>0.321</td>
<td>0.333</td>
<td>0.339</td>
<td>0.340</td>
<td>0.343</td>
<td>0.343</td>
</tr>
<tr>
<td>CRF+S+E</td>
<td>0.335</td>
<td>0.339</td>
<td>0.341</td>
<td>0.342</td>
<td>0.341</td>
<td>0.339</td>
</tr>
<tr>
<td>CRF+S+A</td>
<td>0.342</td>
<td>0.342</td>
<td>0.343</td>
<td>0.343</td>
<td>0.343</td>
<td>0.342</td>
</tr>
</tbody>
</table>

### 8.2.5 Discussion

The NCE score results in Table 8.2 highlight the difference between the posterior lattice-based approaches discussed in this work. The first interesting observation that can be made based on these results, is that the CRF-based system exhibits exactly the same performance as the mapped posterior scores. This result is however expected, as the CRF model used to conduct these experiments used only the single posterior-based predictor score as its input features, and the mapping approach was chosen to be exactly the same as the baseline mapping function. This result proves that the proposed CRF models are guaranteed to achieve baseline performance using these features.

In comparing the systems and interpreting the NCE scores, care should be taken to consider the differences in WER for each system. These differences are a natural result of the fact that each approach effectively results in different transcriptions. The time-based posterior method is based on the standard lattice 1-Best transcriptions, while the other approaches produce “consensus” hypotheses. However, the fact that the WER and NCE scores are similar for both the confusion network posteriors and Lattice-based MBR posteriors, suggests these approaches are indeed very similar. This was to be expected as the evaluation in [44] showed a similar trend. The seemingly high NCE scores for the time-based posterior measures may however not be compared directly with the other approaches, as the system WER is significantly different. This is due to the fact that the NCE measure is one which is normalised relative to the underlying system’s baseline performance. As such, the NCE as calculated in the case of time-based posteriors, is relative to the poorer performance of the underlying system, and is therefore biased.
Figure 8.1: DET Curves for CNP Baselines and Parameter Experiments on Eval04

Building on the baseline experiments, the parameter experiments showed that using a given set of input features (confusion network posteriors), the performance of the system may be improved significantly through the use of a more appropriate internal representation of the continuous input features. The scores shown in Table 8.3 show that the uniform occupancy quantisation approach is superior to the uniform segmentation approach. It is also evident that the spline-based method converges to a certain level of performance almost immediately, while the binning techniques approach their optimal performance levels at higher quantisation parameter levels.
The DET curve for a selection of the evaluated systems is shown in Figure 8.1. In the lower right section of the plot, the baseline unmapped posteriors in fact perform better than any other system. This is an effect of the tendency of these posteriors to underestimate the true confidence measure. However, in the upper left quadrant it can be seen that the “CNP S+A 8” system reaches 0% of incorrect words removed, with about 98.75% of the correct words removed. The baseline system reaches this value only when virtually all words are removed at a value greater than 99.9%.

8.3 Alternative Input Features

The Lattice-based posterior scores experimented with in Section 8.2 are in fact already confidence measures in their own right. In order to justify the CRF-based approach to CE, and investigate the effects of various aspects of the modelling framework, relatively straight-forward alternative input features were sought for experimentation. The experiments in this section are not expected to yield systems which show particularly outstanding performance, but rather are expected to illustrate the effectiveness of the CRF approach. These predictor/input features and the process of their estimation will firstly be discussed, before experiments based on this data are presented in the sections which follow.

8.3.1 The N-Best Homogeneity Score

The N-Best Homogeneity score (NBH) is a predictor score which may essentially be interpreted as a simplistic word-level posterior. This score was therefore chosen for experimentation, as clear comparisons could be drawn between this score and the lattice-based posteriors experiment with in Section 8.2.4. The NBH score is essentially a measure of the consistency with which a recogniser selects a given word in the same position over a list of the N-Best transcriptions. The NBH score was therefore generated for the dev04 and eval04 datasets in the following manner:

• For each lattice corresponding to an utterance in the dataset, the SRILM lattice-tool application was used in order to run an A* Search over the
lattice to recover the 1000-Best\textsuperscript{9} transcriptions.

- The resulting N-Best list was aligned with the 1-Best transcription. A custom tool was developed to perform this task using a Levenshtein DP alignment.

- After alignment, the NBH score was calculated by accumulating the number of times the 1-Best word occurred across the aligned N-Best list at a given alignment position.

- Finally, the accumulated statistics were attached to the 1-Best transcription MLF file for subsequent evaluation.

The same process of generating a parallel corpus of scores and correct/incorrect tags was carried out for these NBH datasets as was described in Section 8.2.

\textbf{8.3.2 The LM Back-off Measure}

The Language Model Back-Off (LMBO) measure was mentioned in Section 3.1.1, and was chosen as a predictor feature for the initial experimentation in this work. The motivation for this choice being that this measure is fundamentally different from the NBH score discussed in Section 8.3.1. The LMBO measure is essentially a discrete value which specifies the order of the n-gram likelihood which is assigned by the LM component of the recogniser, to a word at a given position in the 1-Best transcription.

The required information is however not stored in the recognition lattices, and must be obtained directly from the language model used by the recogniser. Furthermore, there is no standard tool which is able to output such information. However, the LPlex tool in HTK is able to output language model perplexities for a word string, given an n-gram back-off LM. Modifications were therefore made to this tool such that it could optionally output the n-gram context for each word in a given word string along with the default likelihood stream. With the fourgram LM used in generating the lattices supplied as a parameter to this tool, the back-off context for words in the 1-Best transcriptions were generated.

\textsuperscript{9}This is the maximal list length, shorter utterances typically contained significantly fewer alternate transcriptions.
8.3.3 Experiments

A new dataset was constructed for the experiments detailed in this section which consists of both the continuous-valued NBH score, and the discrete LMBO measure. Experiments were carried out on both the single-valued NBH dataset, and the new combined dataset. CRF models which implement binning quantisation were used in the evaluation. It is however expected that if spline approximation was used, performance gains in line with that shown in Table 8.3 would be achieved. There is naturally no principled manner in which the LMBO feature may be interpreted as a confidence score directly. As such a baseline metric could not be defined for the NBH+LMBO dataset. The results of exemplar experiments are shown in Table 8.4.

Table 8.4: NCE Scores for Feature Combination Experiments on Eval04

<table>
<thead>
<tr>
<th>Model</th>
<th>NBH</th>
<th>NBH+LMBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.018</td>
<td>-</td>
</tr>
<tr>
<td>Mapped</td>
<td>0.197</td>
<td>-</td>
</tr>
<tr>
<td>CRF+B+E 4</td>
<td>0.211</td>
<td>0.215</td>
</tr>
<tr>
<td>CRF+B+E 8</td>
<td>0.227</td>
<td>0.231</td>
</tr>
</tbody>
</table>

8.3.4 Discussion

As can clearly be seen in Table 8.4, a performance gain may be achieved when combining multiple input features within the CRF framework, even within this relatively restricted experiment.

Although the LMBO and NBH scores are essentially “different” scores, the same language model was used in extracting the n-gram back-off measures as was used in decoding the N-Best hypotheses over which the NBH is calculated. It may therefore be expected that there is some correlation between these two input features, which does not seem to impair the CRF model’s performance in any way. This is a desirable result as it is one of the fundamental principles which has motivated the application of CRFs in this work.
8.4 CRF Feature Function Experiments

A further aspect of the CRF approach which was preliminarily investigated, is that of complex feature functions. The experiments in this section seek to provide an initial investigation into the potential gain which may be achieved through their implementation.

8.4.1 Bigram Features

Complex feature functions which incorporate contextual information from neighbouring input features are of interest. Such features should allow the model to capture some information pertaining to the longer-range dependencies between input features. Bigram feature functions which operate on two neighbouring feature vectors were implemented and investigated in the experiments which follow.

The standard “observation” feature functions were augmented to act on both the current, and previous observations vectors. Implementing such features results in an exponential increase in the number of model parameters to be estimated. However, for bigram features with a relatively low-order quantisation parameter, the increase in time complexity for parameter estimation is not prohibitive.

8.4.2 Experiments

Bigram features were evaluated on the datasets corresponding to the NBH scores as well as the time-based, confusion network and MBR posteriors in order to investigate the effect of these feature functions. The CRF models evaluated used the default binning technique of a uniform occupancy segmentation (B+A in Table 8.3). A selection of the results of these experiments are shown in Table 8.5. The additional parameter in the model description specifies the quantisation parameter used (i.e. +5 indicates that 5 bins/knots were used).
Table 8.5: NCE Scores for Bigram CRF Features on Eval04

<table>
<thead>
<tr>
<th>NCE</th>
<th>Unigram</th>
<th>Bigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBH (B+A+5)</td>
<td>0.222</td>
<td>0.227</td>
</tr>
<tr>
<td>NBH (B+A+8)</td>
<td>0.229</td>
<td>0.236</td>
</tr>
<tr>
<td>NBH (B+A+20)</td>
<td>0.235</td>
<td>0.240</td>
</tr>
<tr>
<td>TBP (B+A+8)</td>
<td>0.361</td>
<td>0.361</td>
</tr>
<tr>
<td>CNP (B+A+8)</td>
<td>0.333</td>
<td>0.332</td>
</tr>
<tr>
<td>MBRP (B+A+6)</td>
<td>0.344</td>
<td>0.345</td>
</tr>
</tbody>
</table>

8.4.3 Discussion

The increase in NCE scores on the NBHOM dataset shown in Table 8.5 is a desirable result, and the positive effect of the longer-range dependency modelling in the input features is clear.

However, the same degree of improvement was not seen on the posterior-based scores. A potential cause for this may be due to the nature of the posterior estimation process in each approach. These approaches carry out clustering and aggregation over arcs in the lattice in order to arrive at a reduced representation of the hypothesis space (e.g. a confusion network). The result being however that the resulting word-level scores are not based on any strict temporal/localised information. It is hypothesised that the CRF classifier may be unable to distinguish between noise and any real information pertaining to longer-range dependencies in sequences of such scores/input features.

Nevertheless, the fact that a performance gain was seen when a more localised measure was considered, suggests that the model was indeed able to capture some longer-range dependencies in the data. This is a promising result, which validates the theoretical motivations given for feature engineering, and their proposed use in future work.
Chapter 9

Conclusions

The problem of Confidence Estimation for LVCSR is investigated in this report. A thorough analysis of the existing literature was carried out. This prompted initial experimentation aimed at investigating the problem to a greater extent, and the validation of certain concepts to ultimately be proposed as part of future work. The conclusions that may be drawn based on this work are the following:

- CE in LVCSR is a challenging task. However, the potential gains that may be achieved through the application of improved confidence measures are substantial. A large number of approaches have therefore been proposed in the literature. The most significant being lattice-based posterior estimation, post classification and the Utterance Verification framework.

- A view of CE which is commonly taken, is one in which it is assumed the information inherent to a given ASR system is the most suitable source of information for estimation of confidence measures. Lattice-based approaches are particularly interesting examples of techniques in which such a view is taken. Furthermore, such approaches represented an elegant in-system solution to the problem, and are the current state-of-the-art for CE.

- Another view of CE is one in which it is assumed that alternative sources of information which are not known/implemented as part of the recognition process are extremely useful in discriminating between words hypothesised correct and incorrectly. If such a view is taken, post-classification techniques
which are able to combine multiple sources of information in a principled manner, are more suitable to the problem.

- A combination of the two afore-mentioned approaches may indeed yield improved systems for CE.

- A crucial aspect of any CE system, is in the treatment of the available data/input features. Adequate representations of the input features, the formulation of useful derived information, and the means through which aspects of these information sources may be leveraged are essential if a successful CE system is to be developed.

- The standard approach taken to CE in which word-level measures of confidence are estimated is useful in a number of tasks. However, a relatively unexplored area of research is in the estimation of such measures over suitably defined “spans” of words or phrases. Such measures would represent an interesting alternative to word-level scores for a number of tasks, particularly those in which the actual system performance is not directly linked to word-level recognition accuracy (e.g. in dialogue systems).

- Many CE techniques employ some form of heuristic approximation in estimating confidence measures, this is particularly true of existing lattice-based approaches. Approaches based on more theoretically sound techniques which are not heuristic in nature would be expected to yield improved CE systems.

Based on these conclusions, a number of viable research areas which have the potential to ultimately yield improved measures of confidence have been identified, and may be proposed for future work.
Chapter 10

Work Plan and Schedule

10.1 Research Directions

In the sections which follow, a description of the proposed research directions for future work will be provided. The content of this section is intended to substantiate the conceptual discussion of future work given in Chapters 5 and 6, and highlight certain important high-level technical aspects of the proposed work.

10.1.1 Lattice-based MBR for CE

The lattice-based MBR technique for estimation of posterior scores which may be interpreted directly as confidence measures was introduced in Section 5.1.1. An implementation of this approach was used as part of this work. The word-level posterior scores generated by this system were subsequently evaluated, with the results in Section 8.2.3 showing a marginal improvement over the confusion network clustering technique. The theoretical foundation of the approach is however sound, and there are significant areas of interest for future work.

The core of the lattice-based MBR algorithm as it pertains to this work, is the definition of the $\gamma$ statistics. These statistics essentially represent a posterior distributions over words, defined at each “word position” in a reference segmentation. The calculation of these statistics relies on a suitably defined loss function. As was mentioned in Section 5.1.1, the loss function used in the MBR
approach need not necessarily be confined to the word-level Levenshtein edit distance, which is suited to word-level optimisation. An extension of the existing approach in which the definition of the loss function is re-defined so as to act over word segments, is a particularly interesting concept for the purposes of this work. Through this generalisation of the loss function, and subsequent adaptation of the reference optimisation algorithm and definition of the $\gamma$ statistics, a measure of segment/phrase level confidence may be estimated. The necessary adaptations to the algorithm may be developed by building on the existing HTK-based implementation of the word-level MBR algorithms of [44].

The approach described here to obtain phrase-level confidence measures is a particularly elegant one, as the estimation process may be carried out directly on the recognition lattice, without the need for additional models. Furthermore, the MBR approach is generally one which has a strong theoretical basis, and it is a well defined framework within which experimentation may be conducted. A discussion on the construction of a suitable evaluation metric will be presented in Section 10.1.6, which concerns the discussion of phrase-based measures explicitly.

10.1.2 CRF-based CE from Lattices

The future work directions described in this section were collectively introduced in Section 5.1. These approaches are broadly concerned with the application of classification-based CE which acts directly on information in recognition lattices.

10.1.2.1 N-Best features

The core of the CRF-based system introduced in Section 5.2.1 has already been implemented as part of this work. However, the approach is to be extended through the future work described here.

Input features derived from information in the lattice which corresponds to the 1-Best transcription have been considered thus far in this work (see Section 8.2 and 3.1.1). This framework may however be expanded upon, with additional, more sophisticated predictor features being generated and experimented with. The outcome of this phase of the research is of paramount importance in subsequent work, as a set of useful input features and an understanding of the
relation between such features will be gained.

The approach may then be extended to the N-Best case. The afore-mentioned features will therefore be accumulated over the N-Best hypotheses, after alignment with the 1-Best transcription. A predictor feature of interest which shall be incorporated into the approach at this stage, is that of the word-level Levenshtein edit tags resulting from the alignment. This feature is analogous to the edit distance calculation carried out within the lattice-based MBR approach discussed in Section 10.1.1. The number of parameters to be estimated will increase dramatically in the N-Best setting however, and due care should therefore be taken to maintain the feasibility of the approach.

The N-Best approach is expected to be more directly comparable with the Lattice-based posterior approaches than the 1-Best approach, as more of the hypothesis space is taken into account. Should the models developed through this part of the work approach the performance of the current state-of-the-art posterior scores, this would be a significant result. A trained post-classifier acting on the N-Best transcriptions represents a CE system with a significantly smaller in-system footprint than a full lattice clustering algorithm executed for each utterance.

### 10.1.2.2 Clustering with Hidden CRFs

The approach introduced in Section 5.2.2 does not rely on an explicit alignment phase. Furthermore, the clustering problem encountered in competing lattice-based approaches is addressed in a principled manner. These facts warrant the implementation of this technique, and a preliminary discussion of the required implementation follows.

Central to this approach, is the use of latent variables in a Hidden CRF model, which should be implemented as an extension of the CRF toolkit. This hidden representation essentially represents a mapping between the input feature vectors and the correct/incorrect tags assigned to the 1-Best transcription. The proposed definition of the input feature vectors in this approach is crucial and calls for further explanation. Firstly, the lattice may be segmented into temporal regions, the span of which being fixed and constrained to be some minimal arc length. The arcs which span each interval may then be considered to be consistent over that
short interval. A feature vector of arc scores may then be constructed for each such interval. The dimensionality of these input feature vectors should however be constrained by a likelihood beam width $\beta$ and a maximum dimensionality.

The input features accumulated over the intervals may theoretically take on any form. One particularly interesting alternative to the arc score is the arc posterior likelihood which may be calculated in a pre-processing step with the forward-backward algorithm. Features such as the overall arc density and the utterance length are also expected to be useful. These features, amongst others, should therefore be investigated as part of the work surrounding this approach.

The method of creating input feature vectors based on a beam width is a general one which may act directly on a recognition lattice without significant pre-processing being required. However, an approach which may indeed yield better results is to generate the input feature vectors from the arcs corresponding to the $N$-Best paths through the lattice. This should also be implemented as an alternative approach.

The approach detailed here is a particularly interesting one, as no heuristic measures of overlap between the arcs need be defined in order to align them with the 1-Best transcription. Furthermore, it has a clear theoretical basis, as a purely probabilistic view of the alignment of groups of informative features with a corresponding confidence tag is taken.

10.1.3 Rich Features

Combining multiple predictor scores within the CRF framework was introduced in Section 6.1.1. As CRF models are particularly well suited to the task of combining multiple input feature streams, this approach warrants further research. A preliminary discussion of the required implementation follows.

Experiments corresponding to the generation, combination and evaluation of multiple predictor feature scores forms part of the work described in Section 10.1.2.1. Building on this lattice-based feature combination approach, input features which are generated from alternative information sources may be investigated. The output of a phone-only decoding of the utterance, mentioned in Section 6.1.1, represents a particularly interesting source of information that
may be investigated. Assuming the lattice of the underlying ASR system has been marked-up with phone labels, the 1-best phone sequence may be aligned with the alternative model phone sequence. The resulting feature streams may then be used as input features for a CRF model. Naturally, these phone-level features may be combined with any of the other so-called “rich features” in estimating confidence scores for the word hypotheses.

The derivation of suitable input features and their implementation within the discriminative CRF framework, which is insensitive to correlations between such input features, is a potentially rich area for research and experimentation. It is therefore hoped that significant improvements in CE performance may be achieved through the implementation of these approaches.

10.1.4 Complex Feature Functions

The benefit of defining arbitrary feature functions to leverage dependencies in the underlying data for CE was discussed in Section 6.1.2. Such feature functions represent a principled way to account for long range and arbitrary dependencies in the data. Furthermore, experiments presented in Section 8.4.1 showed an improvement in NCE accuracy with bigram features alone using an N-Best homogeneity metric predictor feature. Further research into complex feature functions is therefore warranted, and a preliminary discussion of the required implementation is given below.

Building on the afore-mentioned experiments, increasingly complex feature functions may be constructed. A first experiment that of expanding the bigram features to act over all features in the input feature vector, such that high-order, multi-modal input feature models may be evaluated. Feature functions which span longer contexts than the bigram may also be investigated. However, it is expected that the number of parameters to estimate with higher-order contexts will make the approach increasingly less feasible.

Skip-chain features (discussed in Section 4.3.1), are an interesting technique which represents a significantly more compact means through which long range dependencies in the data may be accounted for. A particularly interesting skip-chain structure which may be implemented, is one based on the LMBO score.
Skip edges may be placed between *sequences* of observations which have been estimated from the same order n-gram, for the same number of successive observations. The motivation for this being that long sequences of high-order n-gram contexts are likely to correspond to sequences of correctly hypothesised words. Conversely, long sequences of lower-order n-gram contexts are more likely to consist of incorrectly hypothesised words. This underlying dependency in the data may be captured and leveraged through the implementation of such skip-edge feature functions.

The design of suitable complex feature is another area of research within the proposed framework, which represents a rich area for experimentation and potential for improvement of CE.

### 10.1.5 Hierarchical Representation

The application of a hierarchically structured *modelling* approach to CE was introduced in Section 6.1.3. This representation makes it possible for features defined at multiple levels of abstraction to be elegantly combined within one model. This is expected to be a useful representation of the underlying data for the task of estimating confidence measures, and further research into the approach is therefore warranted. A preliminary discussion of the required implementation is given below.

The approach calls for the development of a Hierarchical CRF implementation as an extension to the existing CRF toolkit, as well as the AIO algorithm necessary for parameter estimation. The fixed hierarchical structure that may be considered initially, is one which makes a hierarchical distinction between suitably defined phone-level and word-level features and feature functions. The phone-only features discussed as part of the approach in Section 10.1.3 are natural candidates for a first implementation. This will also allow direct comparison with the afore-mentioned “flat” system.

More suitable predictor features which are supported by the hierarchical framework may be generated, and complex feature functions which are able to act on these features may be developed as part of the work. Essentially, these tasks relate back to the more general future work described in Sections 10.1.3 and
10.1.6 Phrase-level confidence measures

Confidence estimation at a sub-utterance level above that of words is an area of particular interest for future work (see Section 6.1.4), as it is a particularly novel concept which has not been addressed extensively in the literature. One approach to the problem was described in Section 5.1.1. The discussion here is however concerned with classification-based approaches.

As was mentioned in Section 6.1.4, the concept of a phrase-level confidence measure, and the *phrasal error region* over which it is defined, must be formalised. Once such a *phrasal error region* has been suitably defined, an extension of the NCE score based on similar information-theoretic principles shall be formulated. Given this definition, and an associated scoring technique, work which investigates techniques through which confidence measures may be estimated over these intervals may commence. A preliminary discussion of the implementation of these techniques follows.

A first approach to estimating such confidence measures, may be one which is based on the use of alternative transcriptions from an N-Best list. The concept being that of implicitly representing the “phrases”, or error spans, by using a measure of consensus/similarity defined over the competing hypotheses. Spans of words which consist of *only correct* or *only incorrect* tags across the hypotheses may be considered equivalent. Given the resulting consensus segmentation, a CRF model acting on the features corresponding to the 1-Best transcription may be used to assign confidence scores to the resulting segments. It should be noted that in this initial approach the phrasal error span is defined as those spans over which *all* words are to be considered correct/incorrect.

As an alternative to the implicit segmentation approach already presented, a classification approach which does not require a definition of the segmentation of an utterance into phrases, is the use of Semi-Markov Hierarchical CRFs. These models were briefly described in Section 6.1.4, where it was mentioned that they allow for an arbitrary segmentation of the input features into labels of phrase confidence to be defined *dynamically*. Should this approach to phrase-based CE
ultimately be taken, it shall be implemented as part of the CRF toolkit for subsequent experimentation. The adapted AIO algorithm necessary for parameter estimation in standard hierarchical CRFs may also be applied to these models.

10.1.7 Utterance Verification

The use of CRF models within a UV framework was introduced in Section 6.2. It is hypothesised that the discriminative nature of the CRF models used, and the relative margin between the distributions modelled by each CRF, will result in a UV likelihood ratio which is highly discriminative in assigning accurate confidence measures to word hypotheses. This approach is therefore a candidate for further research, and an initial discussion of the implementation follows.

Within the UV framework, the models corresponding to the null and alternative hypotheses are intended to represent the conditions under which an underlying ASR system may produce correct or alternatively, erroneous transcriptions. CRF models of a similar form to that described throughout the proposed future work may be used as the $\Lambda_w$ and $\Lambda_a$ models. However, the $\Lambda_w$ and $\Lambda_a$ models should naturally be trained on orthogonal data sets which represent the conditions under which the null and alternative hypothesis are valid. Datasets scored in the same manner as discussed throughout this section with SCLite, will be split into positive and negative exemplar subsets, corresponding to correct and incorrect word hypotheses in transcriptions. As there will be significantly fewer negative training samples than positive, it is expected that it will be necessary to combine the negative examples from more than one of the datasets in Table 8.1, or indeed generate more datasets.

The input features/observations used by the models may of course be chosen arbitrarily, and may include the rich features discussed in Section 10.1.3. A first implementation of the approach will only consider features which are related to the 1-Best hypothesis. However, this may subsequently be generalised to the N-Best case, in a similar fashion to that described in Section 10.1.2.1. The CRF models trained on these data sets may then be applied within a LRT framework (see equation (3.5)), to evaluate the difference in evidence under the modelled distributions corresponding to the null and alternate hypotheses. The raw LR
score evaluated in this manner may be interpreted as a measure of confidence. However, a subsequent mapping phase is likely to be necessary in order to yield standard confidence scores in the interval $[0, 1]$. The cubic spline approximation technique previously used in the context of CRF continuous feature functions is proposed as a means through which this mapping may be applied.

### 10.2 Proposed Timeline

A number of research directions were proposed in Section 10.1. However, some of these proposals for future work are of more interest than others, and the research work undertaken should be prioritised accordingly.

The following are considered to be research questions to be investigated as part of the primary research track:

- **Rich input features:** The investigation of suitable predictor features, drawn from lattices and alternative models is a central aspect of the proposed research on which much of the subsequent work relies. It is hoped their application within the probabilistic framework of CRFs will yield significantly improved models for CE.

- **Complex feature functions:** A principled manner through which long-range, arbitrary dependencies may be defined between rich features in order to implicitly model aspects of the data, should prove to be a powerful approach to CE. The work carried out in investigating complex feature functions will also be applicable to all CRF-based approaches investigated.

- **Hidden CRF-based consensus clustering:** A classifier which is able to act directly on a recognition lattice, and derive accurate confidence measures based solely on the available information in the lattice, would represent an elegant CE solution. The sound statistical foundation of the approach, in which confidence measures may be estimated without relying on heuristic techniques, serves as further motivation for research into the approach. It is hoped that such an approach will yield improvements over comparable current state-of-the-art lattice-based systems.
• **Lattice-based phrase-level CE:** The estimation of phrase-level measures of confidence is a particularly challenging research question, for which no significant solutions have been found. This is therefore a potentially rich area of research. The estimation of these measures directly on a lattice is a particularly interesting approach which aligns well with the conceptual task at hand, and is expected to represent a useful solution to the problem.

The following are considered interesting research questions which should be addressed as part of this work if time allows:

• **Discriminative UV:** The application of discriminative models such as CRFs to UV has not been researched extensively, despite the fact that such models are a natural candidate for implementation within such an approach. This is essentially an approach which builds on work considered to be primary research goals, and as such should be the first of the secondary research questions to be addressed.

• **Hierarchical CE:** Hierarchical models build on the complex feature function and input feature aspects of the primary research goals. However, the implementation of a fairly complex model structure is required. It is also unclear whether much will be gained through using a complex hierarchical model to act on features defined at different levels of granularity, over an approach which makes use of a “flat” model with suitably defined feature functions. Nevertheless, once a Hierarchical model has been implemented, there will be no need to explicitly engineer functions for such multi-level features. This approach should therefore be investigated if time allows.

• **CRF-based phrase-level CE:** Should the phrase based approaches investigated in the primary research track not yield significant results, or if time allows, the Hierarchical semi-CRF approach to phrase-level confidence estimation should be investigated. The approach involves the implementation of a highly complex model. However, the challenges present in phrase-based CE play specifically to the modelling strengths of these models.

The primary research questions with the estimated absolute time investment for each are shown in Table 10.1. However, the rich input feature and complex
feature function work is related and will be investigated in parallel. Furthermore, lattice-based clustering and phrase confidence estimation are also related themes, and should be investigated in parallel. The overlap between the investigation of these research questions is illustrated in the Gantt chart (Figure 10.1).

On a more immediate basis, the CRF framework for CE applied to experiments with rich sources of information (in particular the phone-only predictor feature), is expected to constitute a significant research output. This may be presented in a paper for ICASSP 2011, for which the submission deadline is \textbf{October 20, 2010}. 

Table 10.1: Time Schedule for the Primary Research Track

<table>
<thead>
<tr>
<th>Activity</th>
<th>Period (months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rich Input Features</td>
<td>5</td>
</tr>
<tr>
<td>Complex Feature Functions</td>
<td>4</td>
</tr>
<tr>
<td>Lattice-Based phrase-level Measures</td>
<td>5.5</td>
</tr>
<tr>
<td>Hidden CRF-based consensus clustering</td>
<td>4.5</td>
</tr>
<tr>
<td>Final Experimentation</td>
<td>2</td>
</tr>
<tr>
<td>Thesis Write-up</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 10.1: Proposed Work Schedule

Term 4
- Rich Input Features
- Complex Feature Functions

Term 5
- Phrase-level Measures
- Hidden CRF Clustering

Term 6
- Final Experimentation

Term 7
- Final Experimentation

Term 8
- Thesis Write-up

Term 9
- Thesis Write-up
References


REFERENCES


[35] Charles Sutton and A. McCallum. An introduction to conditional random fields for relational learning. in introduction to statistical relational
learning, edited by lise getoor and ben taskar. *Introduction to statistical relational learning*, [x]:93, 2006. 40, 41


