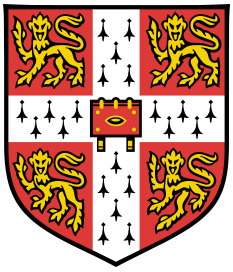
Confidence Estimation for Automatic Speech Recognition Hypotheses

MATTHEW STEPHEN SEIGEL

ST EDMUND'S COLLEGE



Department of Engineering

University of Cambridge

This dissertation is submitted for the degree of Doctor of Philosophy

December 2013

Confidence Estimation for Automatic Speech Recognition Hypotheses

Matthew Stephen Seigel

Abstract

Automatic speech recognition (ASR) systems produce transcriptions for audio which sometimes contain errors. It is useful to know how much confidence may be placed in this output being correct. Confidence estimation is concerned with obtaining scores which quantify this level of confidence. The development and application of a principled, flexible framework using conditional random field (CRF) models for confidence estimation is described.

Errors tend to occur over a number of consecutive words in ASR output. This phenomenon is not typically accounted for in confidence estimation, but is exploited here through the sequential nature of the CRF. A custom CRF framework is developed, making it possible for useful feature functions to be engineered. This framework is extended to support hidden-state CRFs. To inform this confidence estimation model, novel predictor features indicative of the quality of ASR hypotheses are proposed, along with a technique for their extraction from lattices.

The CRF-based approach is used to combine multiple predictor features and estimate confidence scores for words in ASR hypotheses. This yields performance improvements in the normalised cross entropy (NCE) metric of up to 11.4% relative to a strong baseline (using decision trees). The novel application of a hidden-state CRF to this task yields further relative improvements of up to 17.2%. Estimating confidence scores on the sub-word-level is also investigated. Sub-word-level features are combined with word-level features to yield improvements of up to 31.7% relative. The use of a hidden-state CRF for this task yields even larger relative gains of up to 48.6%.

The application of CRFs to estimate keyterm confidence scores for spoken term detection is proposed. Discriminative features for keyterm hypotheses are introduced, as well as a model-based approach to keyterm score normalisation. This approach results in improvements of 26% and 36% relative in the miss rate and false alarm rate at operating points of interest.

The novel task of detecting deletions within ASR output is investigated. The sequential nature of the CRF is exploited to make this possible, such that regions in which deletions occur are modelled. Modelling word confidence and deletion regions simultaneously yields an approach which is capable of detecting deletions.

Overall, the proposed framework for confidence estimation is shown to yield improved confidence estimates. This is important for downstream applications (e.g. dialogue systems, keyterm detection) which make decisions based on these scores, as well as in-system applications (e.g. data selection and adaptation).

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. A large proportion of this thesis has been published, and presented at international conferences. Chapter 7 of this thesis is based on the work surrounding the use of CRFs for word-level confidence estimation (Seigel and Woodland 2011). Chapter 8 is based on the extension of this framework to the sub-word level (Seigel and Woodland 2012), with that work also introducing hidden-state CRFs for confidence estimation used in both Chapters 7 and 8. The application of CRFs to keyterm confidence (Seigel *et al.* 2013) is the basis for Chapter 9.

The length of this thesis including appendices, bibliography, footnotes, tables and equations is 64 787 words. It contains 29 figures and 18 tables.

Acknowledgements

Firstly, I'd like to thank my supervisor, Prof. Phil Woodland, for taking me on as a student, and providing me with extensive support throughout the course of the PhD. I feel privileged to have had the opportunity to be supervised by Phil, and have been able to learn a great deal from his experienced guidance and support. I am grateful for the time Phil set aside for our regular meetings, as this time has been invaluable during the course of my research. I'd also like to thank him for his role in securing funding support for me, and giving me the opportunity to get involved in the related research efforts of the speech group (DARPA AGILE and RATS). On that note, I'd like to thank the Nuance Foundation for their funding support, which has seen me through the latter stages of my PhD.

I'd like to thank Prof. Mark Gales, whom I was fortunate to have the opportunity to work with on the RATS project. Mark's insightful guidance made it possible for me to further my knowledge of various aspects of "real" speech recognition systems. Mark also patiently supported and fostered my work in furthering parts of my research as part of this project, for which I am truly grateful.

My time has been enriched during the PhD through interactions with members of the machine intelligence lab. I've enjoyed having countless lively academic (in addition to many not-so-academic) discussions with members of our office (Juan Pino, Matt Shannon, Rory Waite) and others in the group (Rogier van Dalen, Milica Gašić, Zoi Roupakia). This list is by no means exhaustive, and I would like to thank everyone in the machine intelligence laboratory with whom my time here has overlapped.

I'd like to thank Rogier van Dalen and Milica Gašić explicitly for proof-reading parts of my thesis. I am grateful they were able to take time from their schedules to do so, and appreciated their candid criticism and insightful comments.

I'd like to thank my family (Dad, Ma, Jason and Amber) for their unending support of my endeavours, for encouraging me, and making much of this possible. Last, but certainly not least, I'd like to say a massive thank you to Emily for her patience and support in general, and especially for her "cheerleading" efforts during the final push for finishing up and submitting this thesis.

Contents

Notation

Variables	
$\mathbf{x}$	A vector
$\mathbf{X}$	A general matrix
X	A random variable taking on values
x	A discrete or continuous value

Operators	
$\max \phi(x)$	The maximum value of $\phi(x)$
$\arg\max_x \phi(x)$	The value of x that maximises $\phi(x)$
$\lvert x \rvert$	The absolute value of x

Distributions	
$P(X)$	The probability mass function of a discrete random variable
$p(X)$	The probability density function of a continuous random variable
$\mathcal{L}(\cdot)$	The log likelihood function
$H(\cdot)$	Entropy
$H(\cdot\lvert\cdot)$	Conditional entropy
$P(\cdot\lvert\cdot)$	Conditional probability distribution
$E(\cdot)$	Empirical expectation
$\tilde{E}(\cdot)$	Model expectation
$Z(\cdot)$	Normalisation or partition function
$\mathcal{N}(\mu, \sigma^2)$	The normal/Gaussian distribution with mean μ and variance σ^2
$\delta(x, y)$	The Kronecker delta function, returning 1 when x and y are equal, and 0 otherwise.
$\mathbb{1}$	A constant value function, returning 1

Speech recognition principles and parameters	
$\boldsymbol{W}$	An ASR hypothesis (e.g. 1-Best), consisting of a sequence of words
$\boldsymbol{S}$	An alternative hypothesis
Λ_W	An HMM for word/sub-word W
Θ	A set of model parameters
$b_j(\cdot)$	The HMM output distribution for state j
a_{ij}	The HMM transition probability between state i and j

$\mathbf{q}$	A state sequence
γ	The grammar scaling factor
$p_{ac}(\cdot)$	The acoustic likelihood
$P_{lm}(\cdot)$	The language model probability
$P_{pr}(\cdot)$	The pronunciation probability
G_i	A graphemic sub-word at position i in a word
t_b	The beginning time for a hypothesis or lattice arc
t_e	The end time for a hypothesis or lattice arc
$\mathcal{I}$	A set of word or sub-word lattice arcs which intersect with a hypothesised word or sub-word

Conditional Random Fields

$\mathcal{G}$	A general graph
$\mathcal{V}$	A set of graph vertices
$\mathcal{E}$	A set of graph edges
$\mathcal{N}(v)$	The set of neighbouring vertices for vertex v
f_k	A generic CRF feature function
λ_k	A generic weight or parameter for a CRF feature function
$t(\cdot)$	A CRF transition feature function
$g(\cdot)$	A CRF observation feature function
y, y'	A template for current and previous label values
y, x	A template for current label and observation values
l	A template value for a literal variable
$\mathbf{Y}$	A label sequence
$\mathbf{X}$	An observation sequence
$\mathbf{H}$	A hidden state sequence
$\mathbf{X}[d]_i$	The value of dimension d in the observation sequence at index i

Indices and sizes

t	Time index
i	Sequence index
T	Length of an utterance or sequence

Keyterm detection

K	A keyterm
K'	The best word immediately preceding keyterm K
K''	The best word immediately following keyterm K
$P_{miss}(\cdot)$	Probability of missing keyterms
$P_{fa}(\cdot)$	Probability producing false alarms
N_{hits}	The number of keyterm hits
N_{fa}	The number of keyterm false alarms
N_{true}	The number of true keyterm events in reference
TP	True positives
FA	False alarms
FN	False negatives

TN	True negatives

Confidence estimation and evaluation

P_c	The empirical word accuracy
P_d	The empirical deletion probability
$\hat{P}(W)$	The confidence score for a word W
$\hat{P}(S)$	The confidence score for a sub-word S
$\hat{P}_{del}(W)$	The confidence in a deletion event following word W
c_{ij}	The confidence score for word i in utterance j
α_{ij}	The ideal (1/0) confidence score for word i in utterance j
H_0	A null hypothesis
H_1	An alternative hypothesis
Z_i	The test statistic for a null hypothesis on segment i
μ_Z	The mean of the test statistic Z
σ_Z	The standard deviation of the test statistic Z

Chapter 1

Introduction

Automatic speech recognition (ASR) technology allows us to interact with machines in a natural way by speaking to them. A plethora of interactive applications built on this technology have become increasingly prevalent in recent times. Examples of such applications include those which access and filter content available on the internet (e.g. Siri, Google Now), control a user interface, perform dictation or even manage access to secure resources. Besides these user-driven interactive applications, services which perform large-scale transcription of audio containing speech, and those which flag the occurrence of specific spoken terms in audio, also rely on the same speech recognition technology. The resulting interest in speech recognition technology research has lead to consistent performance improvements over time, to the degree that current state-of-the-art systems achieve impressive levels of accuracy.

Nevertheless, state-of-the-art systems are by no means perfect, and often produce transcripts which contain recognition errors. Sophisticated statistical models of speech and language are typically employed by such systems. Given sufficient amounts of data for a particular domain, the parameters of these models can be estimated reliably using proven machine learning techniques. However, the aforementioned errors arise when using such systems in real-world situations as a result of two main factors. Firstly, the acoustic environment within which the data was captured that is used to estimate the parameters of the acoustic model may be vastly different from that encountered during test. There are often large differences between the training and test acoustic environments. These differences include changes in the noise conditions, speakers and channel distortions. Secondly, assumptions which

are typically made in the underlying statistical models such that their application becomes tractable, are often too restrictive.

Given that some degree of uncertainty in the accuracy of transcriptions output by an ASR system must be assumed, it is valuable to have a measure of just how much confidence should be placed in the output being correct. The utility of such measures is evident when one considers that downstream applications are required to make decisions based on the output generated by the underlying ASR system. For instance, dialogue systems are able to reason about their belief of the user's requested more accurately using such measures, and guide the dialogue accordingly. In semi-automated transcription tasks, audio segments with low levels of confidence may be transcribed manually or discarded (in data selection), with high-confidence transcriptions being accepted.

The process of obtaining such measures of confidence in the hypothesised transcriptions output by ASR systems is known as confidence estimation (CE). The central theme of this thesis is that of improving upon existing techniques for CE to produce more accurate, reliable measures of confidence that may be utilised to improve performance in applications which make use of ASR technology.

The existing literature on CE falls into one of three classes of approaches (Jiang 2005). In the first approach, the confidence measure for a word is defined as being the posterior probability of the transcribed word. These posterior probabilities may be obtained in a variety of ways. They may for example be estimated during 1-Best decoding (Evermann and Woodland 2000*b*; Wessel *et al.* 2001), consensus network (CN) clustering (Mangu *et al.* 2000; 1999), or during minimum Bayes risk (MBR) decoding (Goel and Byrne 2000; Xu *et al.* 2010). The resulting posteriors are however often not accurate measures of confidence in their original form, and tend to overestimate the true probability of the word being correct. Some form of mapping must therefore typically be applied to these scores to yield more accurate confidence measures. The second approach formulates the CE task as a statistical hypothesis testing problem, in which the ratio between the evidence in support of the word being correct is evaluated against the evidence in support of the word being incorrect. This implies that more than one model must be trained, which represents a significant increase in design effort and overall system complexity. The third approach focuses on the implementation of a classifier to estimate the probability that a given word in the transcription is correct, based on a set of informative *predictor features* which capture some underlying characteristics of the ASR system and are related to the confidence in its output.

This approach is the most flexible, as information from multiple different sources may be combined to inform the confidence estimation process. This is the approach adopted in this thesis.

1.1 Thesis outline

In this work, a class of statistical models called conditional random fields (CRFs) are used to classify word and sub-word level units of ASR hypotheses as being correct or incorrect, to produce accurate confidence measures. In Chapter 6, the contributions made in developing the framework for the application of these models to the confidence estimation task are detailed. In Chapter 7, the task of estimating word-level confidence scores for ASR hypotheses is addressed. The task of sub-word-level confidence estimation is described in Chapter 8, with the required extensions to the framework being presented in this chapter. In Chapter 9, the contributions made through exploiting the fact that confidence estimation may, in theory, be performed for any ASR hypothesis is explored. In this chapter, the task of assigning scores to keyterm hypotheses in keyword spotting systems is formulated as one of confidence estimation. In Chapter 10 of this thesis, contributions made in a novel approach whereby deletions can be detected as part of the confidence estimation process are presented.

1.2 A CRF-based framework for Confidence Estimation

A flexible framework for confidence estimation using CRF models is described in Chapter 6 of this thesis. Firstly, a set of techniques for extracting informative predictor features for ASR hypotheses from recognition lattices is detailed. This includes a description of the novel **hypothesis injection** technique, which makes it possible for features to be extracted for arbitrary hypotheses. Conditional random field models have the advantage that they support the use of arbitrary feature functions. This makes it possible for feature functions to be developed which are capable of capturing specific characteristics of the data or task. Standard software implementations are however limited in their flexibility, particularly in terms of engineering specialised feature functions. In order to take full advantage of this aspect of the model, a **flexible CRF toolkit** (called `CRFTK`) was therefore developed as a contribution of this work. An overview of this toolkit is also provided in this part of the thesis. The standard metric used

to evaluate the performance of CE systems (normalised cross entropy (NCE)), while being useful, is specifically suited to word-level evaluation, and can be difficult to interpret. The **utterance-level mean absolute deviation (UMAD)** is introduced here as a metric which measures the performance of CE systems over sequences of words (i.e. sentences or utterances). Finally, a method for assessing the **significance of CE results** is detailed.

1.3 Confidence Estimation at the Word and Sub-word-level

In Chapters 7 and 8, the application of the CRF-based framework to confidence estimation for 1-best and alternative (confusion network) hypotheses in state-of-the-art ASR systems is detailed. The task of performing word-level confidence estimation is considered first. A set of word-level predictor features are proposed and used. Many of these predictor features are continuous in nature. The standard CRF configuration (as specified by the feature functions typically used), are not well suited to continuous input features. As a result, **spline feature functions** (Yu *et al.* 2010), which were previously applied in maximum entropy models, are incorporated into the CRF modelling framework to address this deficiency. The sequential nature of the CRF model represents information pertaining to the dynamics of errors at the word level. There may however be additional structure in these sequences that can be captured by the model. An approach is therefore proposed whereby **hidden states** are incorporated into the CRF model, and applied to word-level CE. This results in further improvements in confidence score accuracy. These gains are attributed to the fact that the internal structure of longer sequences of erroneous or correct words is effectively modelled by the hidden states, with the word-level sequence also being modelled as before in the standard CRF.

Incorporating additional information sources into the model is shown in this work to typically yield improvements in CE performance. However, not all such additional sources are defined at the word-level, and may be defined at a different level of granularity within the ASR system. One example of such information investigated here, is that of the scores output on the sub-word-level by an alternative recogniser (which is not constrained by a language model). Whilst it is shown in this work that such information is useful if converted to a word-level representation, it is reasonable to assume that information is lost through this averaging process. As a result, the task of **confidence estimation at the**

sub-word level using features which are themselves defined at the sub-word-level is investigated. In the direct sub-word confidence estimation task, confidence scores are sought which indicate whether each sub-word is correct within an ASR hypothesis. Novel predictor features which are defined on this level of granularity are proposed. These predictor features are extracted from the underlying system, as well as an alternative sub-word recogniser. The resulting sub-word-level predictor features are combined with those defined on the word-level to yield improvements in confidence score accuracy. These sub-word-level models are also shown to be more powerful when **hidden states** are used to capture further information on the nature of sequences of erroneous or correct sub-word-level units. A similar task is considered, where the goal is to obtain word-level confidence scores, with the confidence estimation model operating on the sub-word level. This is referred to as word-targeted sub-word confidence estimation. Modelling at this level of granularity does however introduce challenges. In particular, information pertaining to the word-level label sequence (for which confidence scores are required), is lost. **Word boundary** feature functions which capture word-level transition information explicitly are therefore proposed. Using these feature functions, word-level performance can be maintained with the sub-word-level features being used to best effect in achieving further gains in performance.

A number of the contributions presented in Chapters 7 and 8 are based on work published as Seigel and Woodland (2011) and Seigel and Woodland (2012).

1.4 Confidence in Keyterms

In Chapter 9 of this thesis, assigning scores to keyterm[1] hypotheses in a spoken term detection (STD) system is formulated as a confidence estimation task. Here, the task is to obtain a measure of confidence in a keyterm having been present in given audio data. This measure is estimated based on information extracted from the underlying speech recogniser. A system which searches for keyterms at the word level was developed as part of this work, and is described in this section. In the literature, an approach commonly used is that of pre-indexing (Miller *et al.* 2007), whereby an index of words and times at which these words may have occurred in the audio is built using the ASR system output. This static index is then queried to determine keyterm detections. The approach taken in this work is similar in

[1]Query terms to be detected shall be referred to as "keyterms" in this work. These keyterms may consist of a single word (i.e. keywords), or multiple words (i.e. key phrases).

the sense that the hypothesis space over which the search takes place is static (i.e. the audio is not re-decoded if the keyterm list changes). However, a set of predictor features are required by the CRF-based system to assign scores to keyterm hypotheses. Rather than extracting features for all occurrences of all words, a more efficient approach is taken, whereby these predictor features are computed once for all detections of keyterms in a given list, rather than once for all possible words. It should also be noted that it is not computationally expensive to extend the index to include new keyterms (and their related predictor features), or indeed to build a new index of this type. Amongst the predictor features that are computed, **contextual posteriors** and **unigram priors** are introduced as features which contribute in the confidence estimation model to yield improved keyterm scores.

In the aforementioned word-level keyterm search, a keyterm event may only be hypothesised (i.e. detected) if it exists in the recognition lattice. These lattices are however a compact representation of the true hypothesis space. A technique which expands the effective hypothesis space, is to perform an additional search at the sub-word-level. Naturally, the set of predictor features extracted for hypotheses from this system are different from those in the word-level system. As the word-level language model scores are not truly applicable on the sub-word level, a predictor feature based on the **posterior score for the acoustic evidence** is introduced. One issue common to STD systems is that of **score normalisation**, which is a concern due to the fact that the scores assigned to different keyterms typically fall into different ranges. This is due to differences in keyterms such as their length and very different language model scores. These keyterm scores must therefore be normalised such that they approach the true probability of a keyterm event having occurred, irrespective of its length, or how likely its constituent words are to occur in language in general. An elegant solution to this problem is proposed in this work, which exploits the power of the CRF model in terms of defining arbitrary feature functions. A set of **literal moment feature functions** are developed which allow the CRF model to learn parameters for the distribution of the predictor features separately for each keyword. The application of the **CRF-based CE framework** to word-level and sub-word-level **keyterm detection systems** in a hybrid configuration are shown to yield improved performance.

The contributions presented in Chapter 9 are based primarily on work published as Seigel *et al.* (2013).

1.5 Confidence in Deletions

In Chapter 10 of this thesis, the novel task of detecting deletions in ASR output as part of the confidence estimation process is proposed and investigated. This task has not previously been addressed to the author's knowledge. This is primarily due to the fact that a standard classification-based approach cannot be used for this task, as features cannot be obtained for a word that is not hypothesised by the underlying ASR system, which would be required to perform classification. However, it is proposed that the sequential nature of the CRF modelling approach taken in this work can be exploited to learn the characteristics of transitioning from a word hypothesised by the ASR system into a region in which one or more deletions may occur. In the classical confidence estimation scenario considered in Chapters 7 and 8, incorrect words are associated with substitution and insertion errors. The definition of errors is extended for the work in this chapter, such that deletions are also accounted for. It is assumed that a deletion can occur following any word in the ASR hypothesis. As such, the models developed for this task simultaneously estimate a measure of confidence in the word being correct, as well as a measure of how probable a deletion region is to occur following the current word. A deletion-based normalised cross entropy (DNCE) score is proposed to evaluate this approach. The transition sequence structure of the CRF models is shown to be crucial for detecting deletions, with a non-sequential approach unsurprisingly shown to be unsuitable for this task.

1.6 Summary of Contributions

The main original contributions of this thesis are the following:

1. The development of a flexible, CRF-based framework for confidence estimation.

2. Advances in the application of CRF models for word-level confidence estimation in speech recognition, with novel applications to sub-word-level confidence estimation and the proposed use of hidden-state CRFs for confidence estimation.

3. The novel use of CRF models to perform discriminative score mapping for keyterm hypotheses in spoken term detection.

4. A novel CRF-based approach which exploits transition structure to simultaneously detect deletions in ASR output, and estimate a confidence score for each word.

Part I

Background

Chapter 2

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 or letter recognition tasks). Large vocabulary tasks are typically defined as those in which vocabulary sizes of over 10000 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 from an arbitrary speaker. State-of-the-art ASR systems tend to employ speaker adaptation techniques (Leggetter and Woodland 1995; Gales and Woodland 1996; Woodland 2001) 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 need to 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 and text corpora is achieved through the application of suitable machine learning techniques. In the sections which follow, a more detailed treatment of the aforementioned principles which are particularly relevant in the context of this thesis 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 ω 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 a class ω in Ω, or $f : \Re^d \rightarrow \Omega$. 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 therefore 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 performing 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 typically 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. One common technique makes use of the mel-scale. This scale approximates the manner in which the human auditory system perceives changes in frequencies. In order to extract a representation for the audio, a Fourier transform is applied to the audio signal. The frequency axis in this spectrum is warped according to the mel scale, before a discrete cosine transform is applied to the mel log powers. The amplitudes in the resulting cepstrum yields the mel frequency cepstral coefficients (MFCCs) (Davis and Mermelstein 1980; Pols 1977). Another approach derives an estimate of the audio spectrum by modifying the short-term spectrum of speech audio, also obtained by taking a Fourier transform of the audio. This modification is applied using transforms related to the psychophysics of hearing, and results in a set of perceptual linear prediction (PLP) features (Hermansky 1990), which have also found widespread use.

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 in Ω 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 frames 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:

$$\hat{W} = \arg\max_{W \in \mathcal{V}} P(W|\mathbf{X}) \tag{2.1}$$

where the term $P(W|\mathbf{X})$ represents the a-posteriori probability of the class or word W. This form of classifier will therefore assign the class $\hat{W}$ 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 $\left(P(A|B) = \frac{P(B|A)P(A)}{P(B)}\right)$ to yield:

$$\hat{W} = \arg\max_{W \in \mathcal{V}} \frac{p(\mathbf{X}|W)P(W)}{p(\mathbf{X})} \propto \arg\max_{W \in \mathcal{V}} p(\mathbf{X}|W)P(W). \tag{2.2}$$

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 $p(\mathbf{X}|W)$ and $P(W)$ are the "true" distributions, a classifier which implements this decision rule is guaranteed to minimise the misclassification rate[1] (Duda *et al.* 2000).

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, $\boldsymbol{W}$, to a feature matrix $\mathbf{X}$ of observations, equation (2.2) may be restated as:

$$\widehat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} \frac{p(\mathbf{X}|\boldsymbol{W})P(\boldsymbol{W})}{p(\mathbf{X})} \propto \arg\max_{\boldsymbol{W}} p(\mathbf{X}|\boldsymbol{W})P(\boldsymbol{W}). \tag{2.3}$$

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. The resulting distributions are however approximations to the "true" distributions. There are two ways in which these are approximate. Firstly, the forms of the models themselves make use of simplifying assumptions to make them tractable. Secondly, the parameters estimated in training these models are also approximate. The optimal classification rule technically holds under the assumption of model correctness. This assumption does not truly apply, due to the aforementioned approximate nature of the modelling techniques used. However, the models are typically assumed to be correct despite this fact.

2.3 Acoustic Modelling

The class-conditional probability distribution $p(\mathbf{X}|\boldsymbol{W})$, or acoustic model (AM), is typically estimated from a parallel training corpus of speech segments or utterances and the associated transcriptions. This is not a trivial task. One key problem is the fact that the length of the observation sequence is not fixed, and is of variable length. There is also an unknown alignment of frame-level observations with the words they correspond to in $\boldsymbol{W}$. Furthermore, the high-dimensional form of the observation matrix $\mathbf{X}$ makes direct estimation of the conditional likelihood intractable. A common solution to these challenges is to model the joint probability $p(\mathbf{X}, \boldsymbol{W})$, using a set of parametric models of word

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

production having parameters θ. Within 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 (or chain) models a sequence of random variables which represent discrete states. It assumes the Markov property, which means that the distribution over the state for the next time-step is dependent only on the current state. A transition is therefore made at each time step t from the current state i to a next state j, dependent only on the current state i. A related model is one in which at the same time as making this transition, an output symbol $\mathbf{X}_t$ is generated from the output distribution $b_j(\mathbf{X}_t)$ of the new state. This output symbol is typically a vector of continuous variables. The transitions are governed by the probabilities a_{ij}. Thus, for a given state sequence $\mathbf{q}$ of length T this yields the following:

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

where the state sequence $\mathbf{q}$ 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.

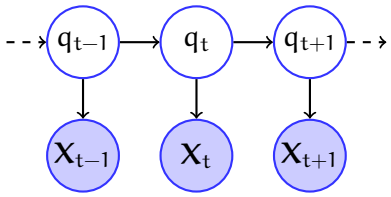


Figure 2.1 *HMM graphical model architecture. The sequence $q_{t-1} \dots q_{t+1}$ is not observed during test (with the corresponding vertices being unshaded). The observed sequence $\mathbf{X}_{t-1} \dots \mathbf{X}_{t+1}$ is depicted using shaded vertices, as it is observed during testing.*

In order to obtain the desired observation likelihood, it is necessary to marginalise out over all possibilities for the hidden state sequence $\mathcal{Q}$ as follows:

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{q} \in \mathcal{Q}} \prod_{t=0}^{T-1} a_{q(t)q(t+1)} b_{q(t)}(\mathbf{X}_t). \tag{2.5}$$

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

$$p(\mathbf{X}|\mathbf{W}) = \prod_{W \in \mathbf{W}} p(\mathbf{X}|\theta_W) \tag{2.6}$$

where θ_W are the parameters of the specific HMM for the word W. The state output distributions $b_j(\cdot)$ are typically modelled using Gaussian mixture models (GMMs). The parameter set θ_W, which includes the transition probabilities and the GMM parameters, must be estimated. These parameters are chosen so as to maximise the likelihood of the model generating the training data. This is known as maximum likelihood (ML) parameter 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 presented thus far has assumed that each HMM represents a linguistic unit at the word level. However, in LVCSR the size of the vocabulary implies 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 phone), for which there is an adequate amount of training data to robustly estimate the necessary parameters. Context-dependent phones are also typically used, which results in an expanded set of of sub-word representations. This once again reduces the amount of data per context-dependent sub-word which can be used to estimate parameters. State-tying (Young and Woodland 1994; Young *et al.* 1994) is a technique which is commonly applied to map the states of similar context-dependent phones to the same equivalence classes such that the amount of data for each state is increased. The sub-word models estimated in this way are ultimately concatenated together based on a pronunciation dictionary to yield more accurate word-level representations.

One significant advance that was made more recently 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. Typical discriminative criteria that are used include maximum mutual information (MMI) (Bahl *et al.* 1986; Valtchev *et al.* 1997; Woodland and Povey 2002), minimum classification error (MCE) (Juang *et al.* 1997; McDermott *et al.* 2007) and minimum phone error (MPE) (Povey and Woodland 2002; Zheng and Stolcke 2005; Gibson 2008).

Significant interest has also been shown in the use of inherently discriminative models for acoustic modelling. These models represent a direct, non generative model of words or phonemes given the observations, as opposed to a joint distribution. This means that such approaches do not suffer as heavily from assumptions made in modelling with respect to the distribution of the data $p(\mathbf{X})$, as the

joint distribution is no longer modelled.

2.4 Language Modelling

The prior distribution $P(\boldsymbol{W})$ is represented by a language model (LM). This distribution is based purely on the word sequence $\boldsymbol{W}$, and in its simplest form may be estimated by obtaining frequency counts from a training corpus of text. Language models which include some word context or history ($n-1$ words in length) in the estimate, are referred to as n-gram language models. 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. Approaches which are typically used to address this issue include the use of models which "back-off" to lower order n-grams for infrequent contexts (Katz 1987), and those which perform statistical smoothing of the n-gram scores (Ney *et al.* 1994; Kneser and Ney 1995).

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 hypotheses for word sequences. Thereafter, a network of HMMs could 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, would include 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, and is often referred to as the 1-best hypothesis.

In all but the simplest of ASR tasks the aforementioned 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:

$$p(\mathbf{X}|\boldsymbol{W}) = \sum_{\mathbf{q}} p(\mathbf{X}, \mathbf{q}|\boldsymbol{W}) \tag{2.7}$$

$$\approx \max_{\mathbf{q}} p(\mathbf{X}, \mathbf{q}|\boldsymbol{W}) \tag{2.8}$$

where the state sequence $\mathbf{q}$ 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 $\hat{\mathbf{q}}$ 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 $\hat{\mathbf{q}}$ need be recovered from the HMM network.

The search space may be constrained further in LVCSR tasks by carrying out hypothesis pruning (also known as beam search). This functions in such a manner that at every time instant, the decoder essentially disregards state sequences which fall below a certain likelihood threshold or *beam width*.

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

$$f(\boldsymbol{W}) = p(\mathbf{X}, \hat{\mathbf{q}}|\boldsymbol{W})P(\boldsymbol{W})^{\gamma}\rho^{\|\boldsymbol{W}\|}. \tag{2.9}$$

2.6 Lattices

A lattice is a compact graphical structure output by a speech recogniser, which represents the hypothesis space explored during a recognition pass for some speech audio data. Considering a token-passing approach for ASR decoding (Young *et al.* 1989), these lattices are produced by propagating multiple tokens corresponding to hypotheses in addition to that which yields the best score after every word. Structurally, a lattice is essentially a directed, acyclic graph consisting of collections of nodes (vertices) and arcs (edges). Each word arc corresponds to a word hypothesised by the ASR system. The arcs also encode the acoustic and language model likelihoods for the word, pronunciation variant

information, as well as the identity of the start and end node. Each node corresponds to a particular starting time within the audio. Each such node may be connected to a number of preceding or following arcs, which imposes the required structure. A single word hypothesis is fully specified using the information encoded by the arc and its start/end nodes. An example word lattice produced by recognising audio containing a short utterance is shown in Figure 2.2.

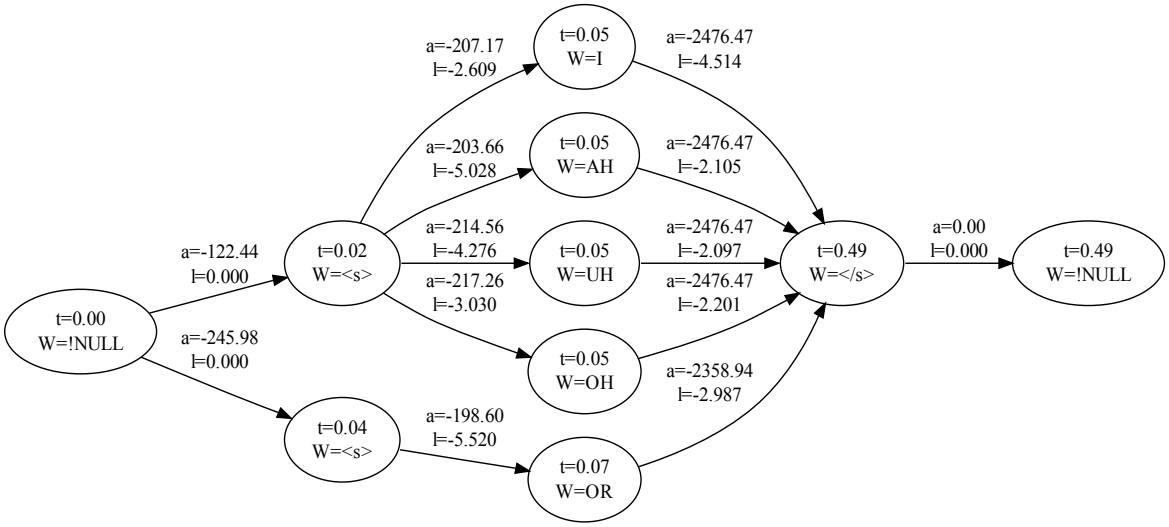


Figure 2.2 *An example word lattice for a short utterance. Arcs and nodes with the associated word identities, timing information as well as acoustic and language model log-likelihoods are shown.*

This representation is very useful as many post-processing techniques can readily access information pertaining to the hypothesised word sequences and their associated temporal information. Additionally, general graph search techniques may be applied subsequent to initial decoding without requiring knowledge of the probabilistic models used by the recogniser when producing the 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 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 occur when a reference word is aligned with a hypothesised word which does not match the reference word.
- Deletion errors, which occur when a reference word cannot be aligned with any word in the hypothesis.
- Insertion errors, which occur when an additional word is present in the hypothesis that cannot be aligned with a suitable word in the reference.

The WER may then be calculated as the resulting total 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 MAP hypothesis or 1-Best transcription). Other metrics are also often used in evaluating LVCSR systems. One such metric is the oracle error rate, which is the lowest WER possible over a list of N competing transcriptions (referred to as the N-Best transcriptions), or over an entire lattice.

Chapter 3

Confidence Estimation in ASR

A measure of confidence in the transcription output by a speech recogniser is an invaluable piece of information in many systems which incorporate an ASR engine. Out-of-vocabulary (OOV) detection (e.g. Young 1994) and keyword spotting techniques (e.g. Wilpon *et al.* 1990) can be based almost entirely on decisions made using such measures of confidence. These applications therefore provided the initial motivation for the development of confidence estimation (CE) techniques. More recently, CE has garnered interest as being useful for many other applications. Examples of such applications include dialogue systems (e.g. Hazen *et al.* 2002), machine translation (e.g. Blatz *et al.* 2004), in system combination (e.g. Evermann and Woodland 2000*a*), and for unsupervised adaptation of acoustic models (e.g. Wallhoff *et al.* 2000).

The problem of estimating accurate measures of confidence in LVCSR systems is however a challenging one, which 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 sufficiently accurate scores. A review of the body of research in the field of CE is presented in the sections which follow.

Confidence estimation approaches may be considered as falling into one of three broad classes (Jiang 2005), each of which will be discussed separately before being compared at the conclusion of the section. As most of the existing literature is concerned with the estimation of word-level confidence measures, this will be the form generally assumed in the discussions which follow.

3.1 Classification Approach

Confidence estimation techniques which fall into this category are concerned with the implementation of a classifier which acts on the output of an ASR system. 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 or incorrect based on a single feature or indeed a set of features. These features may be drawn from numerous sources. One such source is the ASR system itself. The resulting features embody information pertaining to the recognition process at the time a word is hypothesised to have occurred.

3.1.1 Predictor Features

The features used in classifying word hypotheses as being correct or incorrect are commonly known as predictor features. Many such features have been proposed in the literature, a number of which are derived from information which is represented by recognition lattices. Theoretically, an optimal 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 (Kemp and Schaaf 1997) 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 (Sanchis *et al.* 2003) 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.

- Features based on acoustic information. The acoustic likelihood scores normalised per frame or by the number of phones (Pinto and Sitaram 2005), is a representation of the match between the acoustic signal and the hypothesised word. Acoustic stability (Zeppenfeld *et al.* 1997) 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. Each of the K outputs is generated using different values of the Grammar Scaling Factor (GSF) γ. This has the effect of weakening the coupling between the language model and the acoustic model. The premise for this feature is that hypothesised words which are often aligned with the same position under different values for γ are more likely to match the acoustics. In situations where a discriminative acoustic model is used, the posterior probabilities available at the frame or sub-word-level may be used to compute predictor features. For instance, the posteriors output by systems using "acceptor" HMMs may be used to produce such features (Williams and Renals 1997; 1999). A separate sub-word recogniser may also be used to provide complementary acoustic features for confidence estimation (Chase 1997*b*; Cox and Dasmahapatra 2002).

- Language model features. The language model probabilities assigned to words or word sequences were shown to improve CE performance when combined with a system using purely acoustic features (Willett *et al.* 1998). Other, non-probabilistic measures based on the language model have also been proposed. An example of such an LM-based feature is the back-off behaviour (Weintraub *et al.* 1997). The premise for the use of this feature being 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 (Weintraub *et al.* 1997). The premise for the use of such features is that shorter phone sequences and words typically correspond to regions of poor performance of the acoustic model.

- Word-level utterance features. A number of word and utterance-level features based on the word identities have been proposed (e.g. Hazen *et al.* 2002). 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 within 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 Chase (1997*a*). Some of the statistical classifiers which have been used for predictor feature combination include: decision trees (Neti *et al.* 1997), linear models or transforms (Gillick *et al.* 1997; Hazen *et al.* 2002), neural networks (Weintraub *et al.* 1997; Chase 1997*b*), boosting classifiers (Moreno *et al.* 2001), naıve Bayes (Sanchis *et al.* 2012), support vector machines (Zhang and Rudnicky 2001) and maximum entropy models (White *et al.* 2007).

3.1.2 Shortcomings

Combining multiple predictor features has been shown to improve performance over single-feature confidence measures in some cases. Such approaches may only really be successful in improving performance when the individual predictor features are statistically independent. A study in Kemp and Schaaf (1997) showed that most predictor features are however 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 classification framework 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 that of successfully recognising speech, by taking an approach in which statistical pattern recognition techniques and principles are applied to accomplish this. If speech recognisers were perfect, there would indeed be no need for confidence measures, as there would always be absolute confidence assigned to transcriptions output by such a hypothetical

system. The scale of the problem and the assumptions that must be made in typical ASR systems are the primary reasons recognisers produce erroneous transcriptions. However, due to the statistical nature of speech recognition systems, it can be assumed that the recogniser's own belief in the hypotheses it generates is a relatively good indication of the true accuracy for the best transcription it produces. 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 Chapter 2. The MAP decision rule was formulated such that the hypothesised word sequence with the maximal posterior probability is chosen. This posterior probability may indeed be interpreted as a confidence score directly. 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 $p(\mathbf{X})$ would entail summing over all possible hypotheses. This is an intractable task for large vocabulary systems. The result is however that the true posterior distribution is not actually modelled. Posterior probabilities must therefore be estimated by either taking an approach in which certain assumptions are made about the form of the distribution $p(\mathbf{X})$, or otherwise taking an approach in which approximate methods are employed to estimate the distribution explicitly.

In the first type of approach, so-called *filler-based* methods address the problem by using a set of simpler, highly constrained filler models to represent the required distribution. Examples of such techniques include those which make use of all-phone recognition models (Young and Woodland 1994), catch-all models (Kamppari and Hazen 2000), and making an approximation based on the highest word score assigned by the recogniser (Cox and Rose 1996). 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 $p(\mathbf{X})$ has however proved more successful. Recognition lattices are compact representations of the most significant competing hypotheses generated during a recognition pass. As such, the hypotheses in a lattice would contribute the majority of mass to the aforementioned intractable marginalisation over all hypothesis/word sequences to estimate $p(\mathbf{X})$ directly. Computing posteriors over the

lattice is therefore a fairly acceptable approximate method. This does however still represent a computationally intensive task. The problem was be simplified by restricting the summation to the N-best hypotheses in the lattice (Rueber 1997; Wessel *et al.* 1998). 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 sections which follow.

3.2.1 Time-based Posteriors

A technique for estimating word-level posteriors on a complete word lattice was proposed in Evermann and Woodland (2000*b*). It should be noted that this approach is very similar to that detailed for the computation of word posteriors from word graphs in Wessel *et al.* (2001). The first step in the 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 $\mathbf{q}$ through the lattice which includes the word W in word sequence $\boldsymbol{W}$, given the observation sequence $\mathbf{X}$, yields the joint lattice path probability:

$$p(\mathbf{q}, \mathbf{X}, \boldsymbol{W}) = \underbrace{p(\mathbf{X}|\mathbf{q})^{\frac{1}{\gamma}}}_{\text{AM}} \underbrace{P(\boldsymbol{W})}_{\text{LM}} \tag{3.1}$$

where it should be noted that the scaling factor γ[1] 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 of a particular arc (a) can then be estimated by summing over all lattice paths which pass through the arc (the set $\mathbf{Q}_a$):

$$p(a|\mathbf{X}) = \frac{\sum_{q \in \mathbf{Q}_a} p(\mathbf{q}, \mathbf{X})}{p(\mathbf{X})}. \tag{3.2}$$

The summation over lattice paths may be computed efficiently 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 significant importance. The next stage of the technique

[1] The γ value is typically tuned for a given system, and is typically chosen to be in the range of between 6 and 20.

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 (Weintraub 1995). Potential clustering solutions were presented in Wessel *et al.* (1998; 2001), 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 $\mathbf{C}_{\text{sec}}$, in which the sum is taken over all arcs with the same word identity that overlap with the current arc at all. $\mathbf{C}_{\text{med}}$ restricts the sum to those arcs which have the same word identity but overlap with the median time-frame of the current arc. Finally, $\mathbf{C}_{\text{max}}$ is essentially the same as $\mathbf{C}_{\text{sec}}$, with the exception that instead of accumulating the posteriors, only the highest posterior of the overlapping arcs is selected as the posterior score. The $\mathbf{C}_{\text{max}}$ method was shown to yield the best posteriors for use as confidence scores (Wessel *et al.* 2001).

Speech recognition lattices have been discussed in this section as a source of posterior probability-based confidence measures. However, similar methodologies have been proposed for obtaining confidence measures in other tasks. For example, in the domain of machine translation, posterior probabilities based on entire n-grams rather than words may be computed from translation lattices. These posteriors are used as measures of confidence to indicate which regions of the lattices are more likely to be correct. Using this information, fluency of the translations may be improved (Blackwood *et al.* 2010; Gispert *et al.* 2013).

3.2.2 Confusion Network Posteriors

Confusion networks (Mangu *et al.* 2000) are an alternative, highly compact representation of the most likely hypotheses in a lattice. All paths through a confusion network are constrained to pass through all nodes, resulting in a simple linear graphical representation of these hypotheses. The arcs in the confusion network correspond to words (ϵ arcs are added for null or missing words in the hypotheses). The nodes essentially impose a segmentation of the utterance into confusion sets. Figure 3.1 on the following page is an example of a typical confusion network. An algorithm for generating such confusion networks by clustering words into confusion sets is described in Mangu *et al.* (2000) and is summarised below.

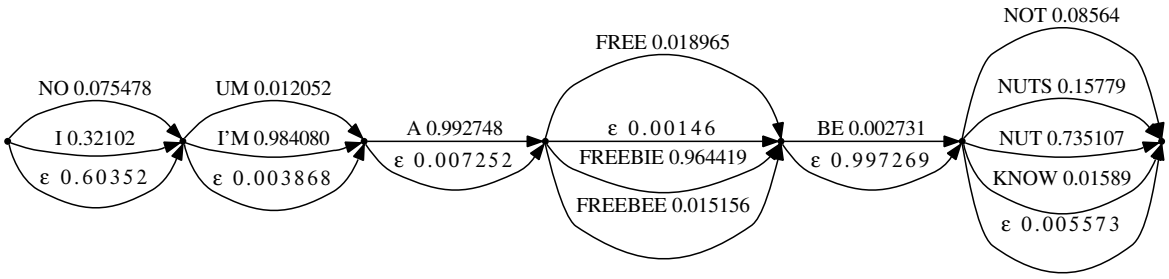


Figure 3.1 *An example confusion network for a short utterance. The* ϵ *edges represent null words.*

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:

$$P(W|t_s, t_e, \mathbf{X}). \tag{3.3}$$

   At this point, each word hypothesis W_i in the sequence $W_0, \ldots, W_L$ of length L represents a singleton cluster, $\mathbf{C}_i$.

3. *Intra-word clustering* is carried out. 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 carried out. 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 them, and a phonetic similarity score weighted by the posterior probability. 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 ϵ 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.

Using this technique for word posterior approximation, an approach in which these posterior probabilities are mapped to confidence scores using a piecewise linear mapping yields a principled approach. Confidence scores estimated in this manner were evaluated for confidence estimation and used for lattice rescoring to minimise Word Error Rate (WER) on a Conversational Telephone Speech (CTS) task (Evermann and Woodland 2000*a*;*b*). In the experimentation carried out in this work, approaches of this type will be used as strong baselines for comparison.

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 Evermann and Woodland (2000*b*)). This necessitates the implementation of an additional post-processing step before the confidence scores can be assigned to words in the transcription. Furthermore, the heuristic nature of the arc clustering and consensus clustering approaches based on overlap and similarity scores is also not ideal, and is a potential source of error in the word posterior probability estimation process. As posterior-based approaches have however generally proven to yield better performance than most alternative single-feature 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 they have been generated by the ASR system. Early work using this approach was in relation to keyword rejection or keyword spotting systems (Rose *et al.* 1995).

3.3.1 *Likelihood Ratio Testing*

A given speech segment $\mathbf{X}$ may be recognised by an ASR system as a particular word W, which is represented by the HMM Λ_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, are then defined as follows:

$$\begin{aligned} H_0 &: \mathbf{X} \text{ was correctly recognised and is generated by HMM } \Lambda_W \\ H_1 &: \mathbf{X} \text{ was incorrectly recognised and is generated by HMM } \Lambda_A \end{aligned} \tag{3.4}$$

where Λ_A is an HMM which corresponds to all alternative word hypotheses that are essentially incorrect. The Neyman-Pearson lemma (Neyman and Pearson 1933) 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 Λ_W and Λ_A respectively, the evidence under each distribution may be evaluated in order to yield a likelihood ratio (LR) expression of the form:

$$\mathrm{LR}(\mathbf{X}, \Lambda_W, \Lambda_A) = \frac{P(\mathbf{X}|\Lambda_W)}{P(\mathbf{X}|\Lambda_A)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau. \tag{3.5}$$

In UV, a thresholding decision (with parameter τ), is applied to this ratio in order to evaluate which hypothesis should be accepted. This LR score may be used as a confidence score directly (Lleida and Rose 1996).

The primary challenge in utterance verification is in the estimation of the alternate model Λ_A. Such models are also know 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 Λ_W (Rose *et al.* 1995; Setlur *et al.* 1996; Rahim 1997). One unifying conclusion which can be drawn from the aforementioned work, 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 Rahim (1997) while the minimum verification error (MVE) criterion was implemented in the work by Setlur *et al.* (1996) and Rose *et al.* (1995). An approach which circumvents the problem of estimating filler models is the exemplar-based technique (Gunawardana *et al.* 1998), whereby the hypothesised word is compared against commonly confused words.

3.3.2 *Shortcomings*

As has already been mentioned, the estimation of the alternate models poses a significant challenge in utterance verification. 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 confidence estimation techniques.

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 (Siu and Gish 1999). The NCE score is a measure of the amount of information which is gained through using the confidence scores under evaluation for each word, rather than a baseline score which is constant over all words - and equal to the overall empirical probability of a word being correct. This baseline score on a dataset of size N, with n correctly hypothesised words is $P_c = \frac{n}{N}$. The evaluation of a CE system therefore requires that the hypothesised word sequence be marked up 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:

$$NCE = \frac{H(\mathbf{C}) - H(\mathbf{C}|\mathbf{X})}{H(\mathbf{C})} \tag{3.6}$$

where $H(\mathbf{C})$ corresponds to the entropy of the tag sequence, and $H(\mathbf{C}|\mathbf{X})$ is the entropy of the confidence score sequence. An NCE score of 0 indicates that the confidence scores under evaluation represents no additional information (and are equivalent to the baseline). Positive scores indicate the confidence measures encode some additional information, and therefore yield an improvement in CE performance. Furthermore, it should be noted that the empirical accuracy P_c is only computed during evaluation, given the reference. This information would not be known by any confidence estimation

[2] This measure is the standard CE evaluation metric employed by the `NIST SCLite` scoring tool

system during test. A confidence estimation system which assigns a constant confidence score that is not equal to the empirical accuracy P_c typically results in negative NCE performance scores.

Given the confidence score output of a system for a set of hypotheses, a decision threshold may be applied to these scores. Above this threshold level, all hypotheses are accepted as having truly occurred, with those below the threshold being deemed to be incorrect. Hypotheses which are accepted, when they are indeed correct, result in *true positives*. Those hypotheses which are rejected with scores below the threshold, when they are indeed incorrect, result in *true negatives*. This decision process also results in two types of errors, *false alarms* and *misses*. False alarms occur when a hypothesis is accepted as being correct, when in fact this hypothesis is incorrect. Conversely, misses occur when a hypothesis is considered to be incorrect, when it is in fact correct. A graphical method sometimes used in the evaluation of confidence measures which has its foundation in signal detection theory is that of receiver operating characteristic (ROC) curves (Egan 1975). In an ROC curve, the false positive rate is plotted against the true positive rate, for a given value of the variable threshold. Each such point on the plot corresponds to a specific *operating point*. 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 random.

A related plot known as the detection error trade-off (DET) curve (Martin *et al.* 1997) is widely used in the literature for comparing confidence measures. In DET plots, the probabilities for each error type (false alarms and misses) are plotted for a range of decision thresholds. Each such threshold and the resulting error probabilities represent a different operating point. A non-linear scale is used for the axes in DET curves, making the plots simpler to interpret than ROC curves. DET curves below the $y = -x + 100$ line indicate system performance which is improved over the random assignment of confidence scores. An ideal system is one in which both error probabilities are 0. As a result, when interpreting DET curves, improvements in performance correspond to curves which are closer to the bottom left corner of the plot (i.e. $p(\text{miss}) = p(\text{false alarm}) = 0$).

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 general 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

Utterance verification techniques represent a framework within which a number of knowledge sources or models can be combined to yield a composite confidence estimation system. This is certainly also true of classification approaches with multiple predictor feature streams. One potential issue with such techniques is however that many single-classifier solutions inherently assume independence of the input features. Lattice-based posterior approaches on the other hand, 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 be argued that such information is the most useful for classifying hypotheses as being correct or incorrect. The crux of the argument being that considerable design effort is expended in developing 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 in a classification approach is however expected to improve confidence estimates. At some point such approaches can stray into the domain of system combination, as the models which represent the additional information sources become increasingly complex. This should be guarded against, as CE is typically intended to be a task that should require less design effort than is required for developing the 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 additional statistical model need be trained. Furthermore, the information required to estimate the posteriors is typically already available as a product of the ASR decoding process. Classification approaches introduce an additional level of complexity as the relevant model needs to be trained, and additional data sources for training need to be constructed. The UV framework necessitates that more than one new model be trained on separate data sets. The challenges encountered in defining a set

of adequate filler/anti-models for UV suggests that this approach is generally 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 order to discover suitable settings for the parameters. Selecting the best combination of predictor features in classification approaches may also represent a significant amount of design effort - unless methods can be found to circumvent this feature selection 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 sub-word (e.g. phone) level if the lattice is marked up with the necessary sub-word timing information. Confidence scores based on such posterior approaches are however only defined at a single level of granularity. Classification approaches yield a principled method of mapping features (e.g. posteriors scores) defined at one level of granularity, to confidence scores defined at a different level of granularity. For example, a system may be trained to effect a mapping of sub-word-level features (such as the posterior probabilities) to word-level confidence scores. Furthermore, considering the fact that multiple information sources can be combined in such classification approaches, it is possible for this information to be defined at multiple different levels of granularity. Utterance verification 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 Longer-span 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-span effects in the data. An example of such an effect is the tendency of recognition errors to occur in runs. It was shown that incorporating contextual in-

formation into a non-sequential approach can improve confidence estimates (Hernandez-Abrego and Marino 2000) by capturing some of this information. However, it is expected that a sequential classifier would inherently be able to model such a phenomenon more appropriately. Lattice-based approaches are not generally able to explicitly model such effects. UV approaches may theoretically be able to model such effects. However, this would entail the estimation of multiple models - each of which may be at least as complex as a 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 (Evermann and Woodland 2000*b*). 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. Another application of lattice-based confidence scores in decoding, is that of confidence-driven lattice cutting and minimum Bayes risk (MBR) decoding (Goel *et al.* 2001). In this approach, lattices are segmented into regions of high and low confidence, before minimum Bayes risk (MBR) decoding is applied to these segments to improve the accuracy of decoded hypotheses. The aforementioned lattice-based techniques are however not compatible with current classification and utterance verification approaches to CE, as they do not operate on lattices directly, and assign scores to a limited number of transcriptions generated by the ASR system.

Another in-system application of confidence scores, is in that of unsupervised acoustic model adaptation (Anastasakos and Balakrishnan 1998; Wallhoff *et al.* 2000). 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 considered more likely to be correct are used for adaptation. Lattice-based methods are more suitable in 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 tighter feedback loop. Methods based on using a classifier on the recognition output are not expected to be as straightforward to implement. This is due to the fact that the model would have to be re-trained using data from the newly-adapted system after each iteration in the adaptation process.

Chapter 4

Sequential Classification with Conditional Random Fields

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 $\mathbf{y}$, given the sequence of observations $\mathbf{x}$:

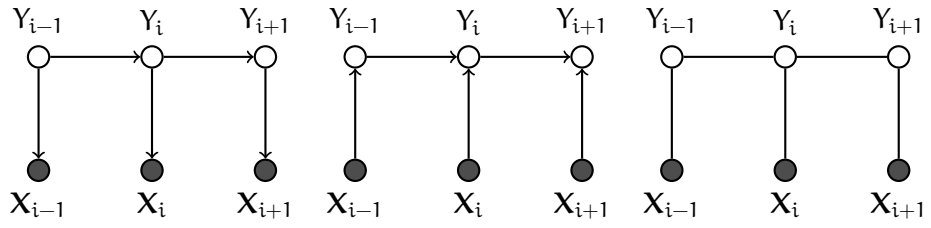
$$\mathbf{y}^{\star}(\mathbf{x}) = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}). \tag{4.1}$$

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 type 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 chapter exploit, is that of a sequential structure in the variables.



(a) HMM **(b)** MEMM **(c)** CRF

Figure 4.1 *Figures illustrating the structure of sequential graphical models. Unshaded nodes depict variables which are not observed during test, with shaded nodes indicating variables observed during test. The directed vertices (arrows) depict dependency relations between random 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 a number of variables. Here, the variables related to the observation sequence, $\mathbf{X}$, and label sequence $\mathbf{Y}$ are considered. A generative classifier may be structured so as to generate a single observation vector $\mathbf{x}$ related to a single/atomic label Y. One approach for classifying sequential data might therefore be to apply such a model at each index/time within a sequence of length T separately. Without a means of representing transitions between labels, such an approach would however not be able to leverage any information pertaining to the sequential nature of the data (i.e. use knowledge of the previous label in the sequence).

Under the assumptions that the current label depends only on the previous label (known as the first-order Markov assumption), and that each observation $\mathbf{X}_i$ 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 chain models, the graphical structure of which is shown in Figure 4.1a. The arcs in this structure are referred to as *edges* in the graphical model nomenclature, and the circles are referred to as *vertices*. Unshaded vertices represent variables which are not observed during test, and the directed edges (i.e. edges depicted as arrows) signify conditional dependence relations between the variables.

These arrows indicate that the variable at the head of the arrow is conditioned on the variable at its origin. Markov chain models factorise the joint distribution over the label and observation sequences $\mathbf{Y}$ and $\mathbf{X}$ 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:

$$p(\mathbf{Y}, \mathbf{X}) = \prod_{i=1}^{n+1} P(Y_i|Y_{i-1})p(\mathbf{X}_i|Y_i). \tag{4.2}$$

The parameters for such models may be 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 generalised form of the Markov chain models discussed previously. 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. A generative model represents a joint distribution, which is not necessary when the problem is in essence a conditional one. The additional complexity in modelling the likelihood of the data, $p(\mathbf{X})$, makes the estimation problem in such generative approaches more challenging (Heigold *et al.* 2012) than in the discriminative case where the class-posteriors $P(\mathbf{Y}|\mathbf{X})$ are used. In addition, generative models typically require that an independence assumption is made such that these approaches remain tractable. This assumption asserts that consecutive observations (or features) are conditionally independent of one another. However, the independence assumption may not always be

valid, and may subsequently impact the accuracy of the resulting model. Discriminative models for sequence classification model the conditional probability distribution of the data, $P(\mathbf{Y}|\mathbf{X})$ directly, and do not make these independence assumptions. As a result, it is possible for the sequence $\mathbf{X}$ to incorporate features which are highly dependent or correlated. Such approaches are therefore interesting alternatives to the generative formalism.

The maximum entropy Markov model (MEMM) (McCallum *et al.* 2000), is a commonly used example of such a discriminative model. It should be noted that in related work which precedes MEMMs, Bourlard and Wellekens (1990) investigated discriminant HMMs as a means of incorporating sequence information into multi-layer perceptrons. In MEMMs, the emission and transition probabilities in traditional Markov chains are replaced with a single transition probability conditioned on the current observation, $P(Y_i|Y_{i-1}, \mathbf{X}_i)$. This represents the probability of transitioning to state Y_i given the previous state Y_{i-1} and the current observation $\mathbf{X}_i$. The resulting model architecture is shown in Figure 4.1b on page 42, in which the shaded vertices represent observed variables. An exponential model is used to represent the distribution $P(Y_i|Y_{i-1}, \mathbf{X}_i)$, 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 aforementioned 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:

$$P^\star(\mathbf{Y}|\mathbf{X}) = \underset{P(\mathbf{Y}|\mathbf{X})\in\mathcal{P}}{\arg\max}\, H(\mathbf{Y}|\mathbf{X}) \tag{4.3}$$

where $H(\mathbf{Y}|\mathbf{X})$ is the conditional entropy of the label sequence, $\mathbf{Y}$, and $\mathcal{P}$ 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 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:

$$E(f_k) = \tilde{E}(f_k). \tag{4.4}$$

Finding the distribution $P^\star(\mathbf{Y}|\mathbf{X})$ in equation (4.3) can then be formulated as a constrained optimisa-

tion 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 (following the notation in McCallum *et al.* (2000)):

$$P_{Y_{i-1}}(Y_i|\mathbf{X}_i) = \frac{1}{Z(\mathbf{X}, Y_{i-1})} \exp\left(\sum_k \lambda_k f_k(\mathbf{X}, Y_i)\right) \tag{4.5}$$

where $P(Y_i|Y_{i-1}, \mathbf{X}_i)$ has been split into $|\mathbf{Y}|$ separately trained distributions of the form $P_{Y_{i-1}}(Y_i|\mathbf{X}_i)$. The normalisation term $Z(\mathbf{X}, Y_{i-1})$ 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 λ_k for each of the feature functions. MEMMs have been successfully applied to a number of tasks. Most notably within the field of natural language processing, for information extraction (IE) and segmentation (e.g. McCallum *et al.* 2000).

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 thesis.

4.2 Linear-Chain CRFs

4.2.1 Definition and Derivation

Conditional random fields (CRFs) were first proposed in Lafferty *et al.* (2001), as a discriminative modelling framework for segmenting and labelling sequence data. Following the definition given in the aforementioned work, the label sequence $\mathbf{Y}$ may be represented by a Markov random field (MRF) using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ is the set of vertices in the graph, and $\mathcal{E}$ the set of connecting edges. The structure of the graph should be such that the label sequence $\mathbf{Y}$ is indexed by the vertices in the graph, $\mathbf{Y} = (\mathbf{Y}_v)_{v\in\mathcal{V}}$. Then $(\mathbf{X}, \mathbf{Y})$ is a CRF if, when conditioned on $\mathbf{X}$, the variables $\mathbf{Y}_v$ obey the Markov property with respect to the graph:

$$P(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = P(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \in \mathcal{N}(v)) \tag{4.6}$$

where $\mathcal{N}(v)$ is the set of neighbouring vertices for vertex $\mathbf{Y}_v$.

The graphical structure of the model may be segmented into a number of sub-graphs called *cliques*. Each clique, $\mathbf{Y}|_c$, 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 (Besag 1974; Lafferty *et al.* 2001)[1], the joint distribution over the label sequence $\mathbf{Y}$ given $\mathbf{X}$ may be factorised into a product of potential functions, ψ_c, acting on each clique. The potential functions essentially perform the task of specifying the relation between the variables which make up the context for which it is defined.

Although the structure of the CRF graph may indeed be arbitrary, attention will be restricted to so-called *linear chain* CRFs in this section (illustrated in Figure 4.1c on page 42). Whilst the figure clearly shows the distinction in the dependence relations between Figures 4.1c on page 42 and 4.1b on page 42, it should be noted that the CRF is also distinct in that it is an exponential model over the entire sequence of states, and is not normalised for each state. Assuming this structure, the cliques are 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:

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

where $\mathbf{Y}|_e$ and $\mathbf{Y}|_v$ correspond to particular cliques, and comprise the set of components of $\mathbf{Y}$ associated with the vertices in subgraphs e and v respectively. Furthermore, $\exp(\lambda_k f_k(e, \mathbf{Y}|_e, \mathbf{X}))$ and $\exp(\mu_k g_k(v, \mathbf{Y}|_v, \mathbf{X}))$ for the *feature functions* f_k and g_k make up the potential functions ψ_c. 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.

[1] This is credited to unpublished work described in Hammersley and Clifford (1971).

- It is possible for the conditional probability of the label sequence to depend on highly correlated complex input features. This is a result of the fact that the model does not require that these features are assumed to be independent, due to its conditional formulation.

- The 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 possible 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 un-normalised fashion. This makes them a good candidate for application to many sequence modelling problems.

4.2.3 *Feature Functions and Input Features*

It is important 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 $\mathbf{X}$). Feature functions are the functions defined by the model which may act on arbitrary combinations of input features (or 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 feature functions 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 feature functions in equation (4.7). These feature functions may be expressed as follows:

$$t_{y',y} = \delta(Y_{i-1}, y')\delta(Y_i, y) = f_{k:k\in\{y,y'\}}(\mathbf{Y}, \mathbf{X}, i) \tag{4.8}$$

$$g_{y,x} = \delta(X_i, x)\delta(Y_i, y) = f_{k:k\in\{y,x\}}(\mathbf{Y}, \mathbf{X}, i) \tag{4.9}$$

where k is an index into a set of K template tuples corresponding to a configuration of state and/or observation values and the Kronecker delta function δ is used to indicate a match between its arguments (a template 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:

$$P_\theta(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_\theta(\mathbf{X})} \exp\left(\sum_{k:k\in\{y',y\}} \lambda_k t_k(\mathbf{Y}, \mathbf{X}) + \sum_{k:k\in\{y,x\}} \lambda_k g_k(\mathbf{Y}, \mathbf{X})\right) \tag{4.10}$$

where θ is the parameter set of the CRF. One difference which is immediately evident between this distribution and that of equation (4.5) for MEMMs, is in the normalisation term $Z_\theta(\mathbf{X})$, which is is not

dependent on the previous state, as in $Z(\mathbf{X}, Y_{i-1})$. The normalisation term in equation (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 $\mathbf{f}$ having K independent features (and K weight parameters $\lambda_1 \dots \lambda_K$).

4.2.4 Parameter Estimation

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

$$P_\theta(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_\theta(\mathbf{X})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(\mathbf{Y}, \mathbf{X})\right). \tag{4.11}$$

The model parameters of this discriminative model are estimated so as to maximise the conditional log likelihood $\mathcal{L}$ that the model assigns to the training data set. It is interesting to note that this approach is similar to that employed in discriminative MMI training of generative HMMs in speech recognition (Nadas 1983; Bahl *et al.* 1986; Valtchev *et al.* 1997; Woodland and Povey 2002). It is also applied in a related discriminative modelling approach (Krogh and Riis 1999), where a globally normalised model is trained based on a hybrid of an HMM and a neural network, called a hidden neural network (HNN). The MMI criterion directly maximises the the posterior of the word sequences, conditioned on the observations. This is also the form of the conditional log likelihood $\mathcal{L}$ for CRF models:

$$\mathcal{L}(\theta) = \sum_{p=1}^{P} \log P(\mathbf{Y}^{(p)}|\mathbf{X}^{(p)}) = \sum_{p=1}^{P} \sum_{i=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(Y_i^{(p)}, Y_{i-1}^{(p)}, \mathbf{X}^{(p)}, i) - \sum_{p=1}^{P} \log Z_\theta(\mathbf{X}^{(p)}) \tag{4.12}$$

where p is a particular training pattern in the dataset $\mathcal{D}$.

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:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \lambda_k} = \underbrace{\sum_{p=1}^{P} \sum_{i=1}^{T} f_k(Y_i^{(p)}, Y_{i-1}^{(p)}, \mathbf{X}^{(p)}, i)}_{\tilde{E}_{f_k}} - \underbrace{\sum_{p=1}^{P} \sum_{i=1}^{T} \sum_{y,y'} f_k(y, y', \mathbf{X}^{(p)}, i) P(y, y' | \mathbf{X}^{(p)})}_{E_{f_k}}. \tag{4.13}$$

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

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 $P(y, y' | \mathbf{X}^{(p)})$ 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 Lafferty *et al.* (2001) gradient descent-based techniques such as generalised iterative scaling (GIS) and improved iterative scaling (IIS) are proposed for estimating the CRF model parameters. An analysis of training algorithms for CRFs found IIS and GIS to be sub-optimal, exhibiting slow convergence rates (Wallach 2003). Furthermore, this work showed IIS and 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 shown 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 thesis will be discussed in the sections which follow.

4.3.1 Higher Order and 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 increasing dramatically as the order (n) of the assumption is increased.

Skip-chain CRFs (Sutton and McCallum 2004) are introduced 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 Sutton and McCallum (2004) these models were proposed for the task of text segmentation, where words having the same identity were linked with skip-chains. The premise being that matching words are likely to be assigned the same class label, and this can be exploited through the use of such features. This approach may however be generalised from performing the word-matching pre-segmentation of the input sequence proposed in that work, to an arbitrary matching function.

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

of the CRF expression:

$$P_\theta(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_\theta(\mathbf{X})} \left\{ \exp\left(\sum_k \lambda_k f_k(\mathbf{Y}, \mathbf{X}) \right) \prod_{(u,v)\in\mathcal{S}} \exp\left(\sum_j \mu_j s f_j(Y_u, Y_v, \mathbf{X}, u, v) \right) \right\} \quad (4.14)$$

where the parameters μ_j correspond to the j skip edge features $sf(\cdot)$, and λ_k parameters correspond to each of 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.

4.3.2 Hidden-state CRFs

The task of a simple pattern classifier is to map a single observation vector $\mathbf{x}$ to a corresponding single class label Y. However, in many sequential classification problems, a variable number of observations in a sequence of observations $\mathbf{X}$ may be associated with a single label Y, with this relationship corresponding to some "hidden" sub-structure in the problem. This hidden structure is not observed in the training or test data, but may be modelled through a *hidden/latent variable* representation. Hidden CRF (HCRF) models (Quattoni *et al.* 2004; Gunawardana *et al.* 2005; Wang *et al.* 2006) are an extension of CRFs which incorporate this type of hidden state-space representation, by augmenting the model with a sequence of hidden-state variables $\mathbf{H}$. Figure 4.2 on the facing page shows a graphical representation of HCRF models. The hidden variables correspond to the sequence $\mathbf{H}$ in the figure. The dotted edges illustrate that the hidden variables $\mathbf{H}$ may also act on longer-range dependencies between input features.

In a more general sequential classification setting, transitions between successive labels may also be captured within the hidden-state CRF model. The resulting model assigns a probability to a label sequence $\mathbf{Y}$ (as opposed to a single label), given the observation sequence $\mathbf{X}$, and does so by marginalising out over all possible hidden state sequences $\mathcal{H}$. This more general class of hidden-state CRF models, sometimes referred to as a latent-dynamic CRF (LDCRF) (Morency *et al.* 2007), is the hidden-state CRF model explored within the context of this work. Figure 4.3 on the next page shows a graphical representation of such a model.

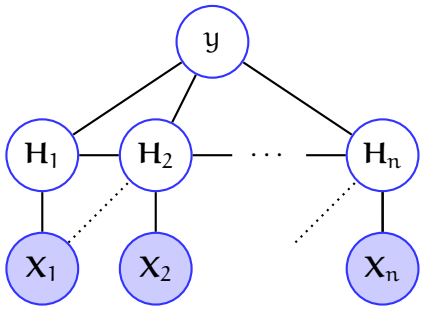


Figure 4.2 *Graphical structure representation for HCRF models, which classify a sequence of observations* $\mathbf{X}_1 \dots \mathbf{X}_n$ *as belonging to a single label* Y*, with associated hidden-state sequence* $\mathbf{H}_1 \dots \mathbf{H}_n$*. The variables observed during test* ($\mathbf{X}_1 \dots \mathbf{X}_n$) *are depicted using shaded vertices, with variables unobserved during test* ($Y, \mathbf{H}_1 \dots \mathbf{H}_n$) *depicted with unshaded vertices.*

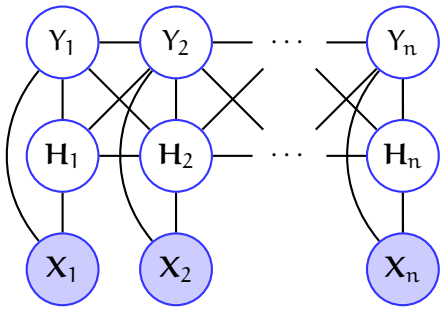


Figure 4.3 *Graphical structure representation for more general hidden-state CRF models, which capture transitions between successive labels in a sequence of output labels* $Y_1 \dots Y_n$*. The variables observed during test* ($\mathbf{X}_1 \dots \mathbf{X}_n$) *are depicted using shaded vertices, with variables unobserved during test* ($Y, \mathbf{H}_1 \dots \mathbf{H}_n$) *depicted with unshaded vertices.*

The HCRF model defines a joint conditional probability of the label sequence $\mathbf{Y}$, and a hidden state sequence $\mathbf{H}$, given the observation sequence $\mathbf{X}$ as $P(\mathbf{Y}, \mathbf{H}|\mathbf{X})$. 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 $\mathcal{H}$. This yields the desired posterior probability of the label sequence $\mathbf{Y}$ as represented by a HCRF model with parameters θ:

$$P_\theta(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{H} \in \mathcal{H}} P_\theta(\mathbf{Y}, \mathbf{H}|\mathbf{X}) \propto \sum_{\mathbf{H} \in \mathcal{H}} \exp\left(\sum_k \lambda_k f_k(\mathbf{Y}, \mathbf{H}, \mathbf{X}) \right). \tag{4.15}$$

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 θ 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 Sung and Jurafsky (2009), 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.

Chapter 5

Keyterm Detection

The internet is an abundant source of information. However, this resource is only useful if it can be queried in such a manner that information on a specific topic may be extracted from it in a meaningful way. This represents a significant challenge, as the search must take place over vast amounts of data. Techniques for text-based information retrieval (IR) from internet web pages have been researched extensively in addressing this challenge. Such techniques are utilised in modern search engines, which are able to return accurate results for a given typed query almost immediately, drawing from the information stored in a massive collection of text.

In recent times, there has been a dramatic increase in the volume of content containing speech audio being generated and made available both on the internet and through other media channels. This is due, in part, to the fact that it is no longer prohibitively expensive to store such data, and broadband internet connections are commonplace. Analogous to the case of text on internet web pages, there is a need for this data to be searchable, as it represents a rich source of information. In addition to this public-domain setting, other tasks such as that of automatic surveillance and broadcast monitoring face similar challenges.

Speech recognition technology is capable of providing the tools necessary for automatically detecting occurrences of specific words or phrases in recorded speech. This motivated the establishment of a track at the National Institute of Standards and Technology (NIST) text retrieval conference (TREC), which fostered a great deal of research in the field of keyterm detection (e.g. Garofolo *et al.* 2000). The majority of work resulting from these evaluations are based on a two-stage approach. The first phase

involves carrying out automatic speech recognition for a given audio recording. Given the text-based output from the first phase, known IR techniques for search are applied in the second phase. This approach asserts the assumption that the transcriptions output by the recogniser are entirely correct. This is however not the case, as ASR systems are prone to errors, and there is always uncertainty in the output due to their statistical nature. It is therefore vitally important that this uncertainty be accounted for (and indeed exploited for benefit), when determining whether or not a given word or phrase may actually exist in an audio recording. The level of certainty or confidence in a query term discovered automatically through this process is therefore typically measured. Applying a threshold to this confidence measure, a decision may be made as to whether or not the query term should be accepted as having truly occurred in the audio.

In this section, the literature which addresses the task of automatically detecting the occurrence of keyterms in audio recordings containing speech will be introduced. The first part will detail the methods used to carry out the search phase. The second part is of more relevance for this thesis, and will focus on the process of computing confidence scores for the keyterm occurrences detected during the search phase.

5.1 Keyterm Search

The task of detecting keyterms within audio recordings has been researched for quite some time, under a number of different titles. Two dimensions along which systems which address this problem are differentiated, are in the length of the keyterms considered, and whether the final list of keyterms is known when the system is built. Systems which perform keyword spotting (KWS) traditionally focus on detecting single-word terms. At the time this nomenclature was prevalent, it was typically assumed that the keyterms were known in advance and did not change. Systems which perform spoken term detection (STD) extend the search for terms to multi-word key phrases and also tend to assume that the list of keyterms to detect is either not known when the audio is first processed, or may be dynamic in nature. Spoken document retrieval (SDR) is a similar task, with the exception being in the application. In SDR, the aim is to return multimedia documents which are relevant for a particular spoken query. The presence or absence of keyterms within the document may be related to the relevance of that

document, but the occurrence of these keyterms is not necessarily truly indicative of the relevance of the document, and is therefore not part of the evaluation for this task. The various approaches to keyterm detection detailed in the literature are summarised in the sections which follow.

5.1.1 Keyterm Classification

The approach taken in the earliest work on keyterm spotting is based on template-matching (Bridle 1973). A sliding window is applied to frames of the speech audio when searching for keyterms. As was proposed for speech recognition in Sakoe and Chiba (1978), the dynamic time warping (DTW) algorithm is applied to match the feature vectors for these frames to template feature vectors for the keyterms. This process results in a number of hypotheses being generated of times at which keyterms are likely to have occurred in the audio. The DTW algorithm may also be applied at the output layer of a neural network model to map from sub-word-level state activations to the keyterm labels (Zeppenfeld and Waibel 1992).

Keyterm classification systems were shown to be quite useful for isolated keyword spotting (Bridle 1973). However, a substantial issue in these approaches is that non-keyterms are not explicitly represented within the process. This implies that the evidence in support of a keyterm occurrence is effectively only evaluated against the evidence for other keyterms and not for non-keyterms. This may be accounted for implicitly to some extent, through the application of a suitable threshold on the distance measure computed between the audio and the template, below which potential occurrences of keyterms are disregarded. However, this approach remains less suitable when keyterms are to be detected within continuous speech, which includes a considerable amount of non-keyterm content. This deficiency is addressed within the same template-matching framework in Christiansen and Rushforth (1977), where the keyterm templates are combined to produce a single, "background" template. The use of additional "filler" templates as well as keyterm and non-keyterm templates was proposed in Higgins and Wohlford (1985).

5.1.2 *Acoustic Keyterm Detection*

Automatic speech recognition systems address the problem of recognising continuous speech. Keyterm detection approaches therefore evolved in step with advances in speech recognition research, as the need for the technology to be applied to continuous speech arose. The first ASR-based approach is introduced in this section, with further examples of work based on this technique being detailed thereafter.

Modern ASR systems often model the acoustic information in speech using a set of hidden Markov models (HMMs), where one such HMM corresponds to a word or sub-word unit in the recogniser's vocabulary. Acoustic keyterm detection (Rohlicek *et al.* 1989; Wilpon *et al.* 1989) is based on the concept of acoustic modelling, and can be viewed as a special case of a full ASR system. Instead of making use of a large recognition vocabulary, only keyterms are considered part of the vocabulary. A whole-word HMM is used to model each keyterm separately and a single "background" HMM is used for all non-keyword speech. A word-loop network which includes separate parallel paths for each keyword and a single path for all non-keywords is generated. During recognition, the Viterbi algorithm is applied using the word-loop network and the relevant whole-word HMMs to find the most likely sequence of words (1-best transcription) for the audio. These 1-best transcriptions are processed to generate an *index*[1]. The inverted index is essentially a look-up table of word identities, each of which is associated with a list of times at which it is hypothesised by the system as having occurred in the audio. Hypotheses (detections) for a given keyterm are generated by looking up the entry for the keyterm in the table.

Unconstrained vocabulary keyterm detection is concerned with the task of detecting keyterms in audio, when the list of keyterms may be altered during test. Applications of a dynamic nature which call for such approaches include surveillance, broadcast news monitoring and voice/video mail retrieval. Acoustic keyterm detection systems which employ whole-word HMMs and search within the 1-Best transcriptions do not scale to this task, as the addition of new keyterms implies that new models must be trained. The approach detailed in Brown *et al.* (1996) addresses this need. This is achieved by using phone-level HMMs and generating phone-level recognition lattices. These lattices are scanned

[1]This term is also common within the information retrieval (IR) literature, used in relation to document retrieval. This is also sometimes referred to as a "posting list".

to detect keyterm occurrences, signified by the presence of the required sequence of phones. Provided a new keyterm can be represented using the set of phones used by the system, it can be detected using this approach.

One challenge faced in acoustic keyterm detection, is that of determining the best way to build an accurate model of non-keyterm (i.e. extraneous) speech. Various forms of models have been investigated to represent extraneous speech, such as garbage, filler and background models (Wilpon *et al.* 1990; Rose and Paul 1990; Lleida *et al.* 1993). Instead of modelling all non-keyterm speech, the approach taken by Rahim *et al.* (1995) makes use of anti-models for each keyword, which are trained using data for all other keyterms. The task of training good models of extraneous speech is however not trivial, as there is a significant amount of overlap in the data used to train the keyterm models, and that used to train the extraneous speech models. As a result, such systems are sometimes unable to accurately distinguish between keyterms and similar-sounding non-keyterms.

5.1.3 Keyterm Search in ASR System Output

The techniques introduced in Section 5.1.2 use principles from ASR to develop specialised (yet constrained) systems which only recognise keyterms. State-of-the art large vocabulary speech recognition (LVCSR) systems are however capable of producing transcriptions that are accurate to a high degree. This is particularly true for domains in which a sufficient amount of training data is available. The approaches discussed in this section exploit the performance of modern LVCSR systems, by implementing keyterm detection as a post-processing step using the output of an LVCSR system. In early work taking this approach, the (low error rate) 1-Best transcriptions output by the recogniser are used to build an inverted index. This index is subsequently queried to return a list of detection times for the relevant keyterms.

Typically, LVCSR systems make use of sub-word HMMs as opposed to whole-word HMMs. This fact can be exploited to improve the flexibility of keyterm detection systems built using such recognisers, making them capable of adapting to changes in the keyterm list. Provided new keyterms can be represented by concatenating sub-word units used by the system, these can be added to the ASR system lexicon, and should also be included in the language model (if not already present). Thereafter, occurrences of these new keyterms may be detected. Whilst it is not necessary to train new acoustic

models using this approach, the audio does however need to be re-decoded before a new index may be built.

There are however two issues in taking the approach detailed thus far. Firstly, a relatively low number of detections are typically generated. This is due to the fact that a keyterm may only be detected if it exists in the single best transcription for a given audio segment (which is a highly constrained hypothesis space). Secondly, it is not possible to detect keyterms which are out-of-vocabulary (OOV). In many applications, a significant number of OOV queries are typical. For instance, in an audio search engine application which indexes a number of news and talk shows (Logan *et al.* 1996), this figure was found to be approximately 12% of all queries. This is largely attributed to named entities, which are often not present in the ASR vocabulary, and tend to change over time. A detailed account of the effects of OOV terms in SDR is presented in Woodland *et al.* (2000), where it was also found that more sophisticated IR techniques can go some way towards improving performance where OOV keyterms are concerned.

Recognition lattices output by an ASR system represent many alternate hypotheses in parallel. It is therefore possible to generate significantly more keyterm detections by building the search index from the word-level recognition lattices rather than from the 1-best transcriptions (Saraclar and Sproat 2004; Miller *et al.* 2007; Vergyri *et al.* 2006). Before indexing takes place, the lattices may also be processed to produce confusion networks (CNs) (Hansen *et al.* 2005), or more convenient representations (which include score information), such as position specific posterior lattices (Chelba and Acero 2005).

An elegant method which addresses the OOV detection problem, similar to that taken in Brown *et al.* (1996) with phone lattices for acoustic keyterm detection, involves generating a sub-word-level index. This index may be built using data obtained in a number of different ways, such as decoding at the sub-word level (Logan *et al.* 2002), converting existing word lattices to equivalent sub-word representations (Witbrock and Hauptmann 1997; Thambiratnam and Sridharan 2007) or using hybrid language models (Parlak and Saraclar 2008). Searching for keyterms not present in the ASR vocabulary is then achieved by querying the sub-word-level index for its constituent sub-word-units. Constraints can be applied to discover valid sequences of sub-word units which occur in the correct order and are sufficiently close together in time to be considered a single keyterm detection.

All the techniques described in this section are based on building inverted indices (effectively look-

up tables) by processing the output of an ASR system. Such systems perform admirably in terms of detecting keyterm occurrences. However, the indexing and search phases can be slow and generally cumbersome to implement, particularly when there is a large amount of audio data to be processed. Weighted automata are proposed as useful tools for text and speech processing in Mohri (1996). This technology may be applied to keyterm search, whereby the audio is effectively indexed as a finite state transducer (FST) and the keyterm or query is represented as a finite state acceptor (FSA). Search is then carried out by performing a composition operation on the source audio FST and the search term FSA. This yields an FST which contains all possible matches of the search term in the indexed audio. The resulting output is generated in time linear to the sum of the number of search terms present in the audio and the length of the search term. This approach has been adopted for spoken utterance retrieval and shown to yield a more general, flexible framework capable of representing arbitrary keyterms that is also highly efficient (Allauzen *et al.* 2004; Parlak and Saraclar 2008; Can and Saraclar 2011).

5.2 Scores for Keyterm Hypotheses

Being able to detect (i.e. hypothesise) the times at which keyterms may have occurred in audio which contains speech is not sufficient for most applications, as not all such hypotheses are equally likely to have truly occurred in the audio. This is particularly true of systems in which a large number of hypotheses are generated. Examples of such systems are those which are tuned so as to reduce the number of missed keyterm events, and those capable of generating hypotheses for OOV terms by clustering occurrences of sub-word units. STD systems therefore typically generate scores for every hypothesised keyterm occurrence. These scores are essentially a measure of how likely it is that the keyterm actually occurred. In this work, this task is approached as a confidence estimation problem, and these scores will therefore be referred to as confidence scores. The real utility of these scores is that they make it possible for system designers to reach a wide spectrum of different operating points. These are set by applying a threshold to the confidence score of keyterm hypotheses. This results in the output of the system effectively being filtered such that only hypotheses with scores over the desired threshold will be present in the output and accepted as corresponding to "true" occurrences. This subset of the overall output will include a certain number of keyword "hits" (correctly hypothesised concurrences), misses

(keywords present in the audio that are not hypothesised), and false alarms (keywords hypothesised to have occurred when they are in fact not in the audio). These counts specify the system operating point, which may be tuned to meet the goals of a particular application. Obtaining accurate confidence scores for keyterm hypotheses is therefore a crucial component of any STD system, having a direct impact on performance. The following sections will summarise the existing literature in which the estimation of these scores is addressed.

5.2.1 Keyterm Posterior Scores

The majority of STD systems make use of some form of posterior probability score for keyterm hypotheses. For a keyterm hypothesis (detection) d, in which a given given keyterm K is hypothesised to have occurred at time t, the posterior score $c(d)$ may therefore be formulated generally as follows:

$$c(d) = P(K|t, \mathbf{O})$$

where $\mathbf{O}$ is the observation sequence. These posterior scores are typically estimated during a standard forwards-backwards pass over the recognition lattices, taking account of the language and acoustic model scores. The use of alternative forms of this posterior based on distilled representations of the lattices such as confusion networks (Mangu *et al.* 1999) or position-specific posterior lattices (Chelba and Acero 2005) have also been proposed.

Posterior scores of the aforementioned form have a straightforward definition when considering single-word keyterms that are in the vocabulary of the recogniser. For multi-word keyterms consisting solely of in-vocabulary words, the overall score is typically calculated by either averaging, multiplying or taking the minimum of the individual posterior scores.

However, in systems which are capable of generating OOV hypotheses, estimating scores for such OOV terms is more challenging. For instance, if approximate string matching is used to generate OOV keyterm hypotheses (Miller *et al.* 2007), there is no sensible score which is analogous to the posterior. Here, hypotheses are therefore accepted if they are within a given term-specific edit distance of the original keyterm.

Estimating keyterm posterior scores in systems which make use of sub-word-level indices may be achieved in a number of different ways. One approach is to estimate the score as a function of the

overlap of the constituent sub-word units of a keyterm (Mamou *et al.* 2007). In systems which perform sub-word-level decoding, local likelihood ratios may also be computed (James 1996). In this approach, a cumulative score based on the likelihood scores of the arcs in a path corresponding to sub-word units of a keyterm is calculated during a forward-backward pass over the sub-word lattices. The ratio of this quantity with respect to that of the 1-best (maximum likelihood) path over the same time interval may be taken as an estimate of the posterior probability of the keyterm. In sub-word-based position-specific posterior lattices (Pan *et al.* 2007), it is assumed that the language model probability for a sub-word unit is equal to that of the overall word, and that the only potential forward path for a sub-word unit in a particular word arc is through the remaining sub-words in that same word arc. Under these assumptions, the posterior probability for a sub-word simplifies to be equal to the posterior probability of the overall word in which it occurs. The posterior probability of the overall keyterm hypothesis is then taken as the normalised sum of the posteriors of all words in which the constituent sub-words of a keyterm hypothesis exist. Whilst not applied within STD, a generalisation of the usual word posterior formulation to the sub-word-level is presented in Lo *et al.* (2004). In this approach the posterior of a sub-word unit is calculated using the actual sub-word acoustic likelihood obtained during decoding. However, it is still assumed that the same language model score applies to all constituent sub-words of a word.

5.2.2 Score Normalisation

The keyterm posteriors described in Section 5.2.1 are useful measures of confidence for keyterm detections. These scores are however *term independent* in that they imply the assumption for decision making that scores of the same or similar value for different keyterms may be treated equally, and that they effectively represent the same level of confidence in the keyterm having occurred. This assumption is however not valid. Differences in the characteristics of the keyterms such as length, frequency of occurrence, and language model scores, have the impact that confidence scores tend to fall into different ranges for different keyterms.

One approach which addresses the term independence issue, is to filter the system output to accept or reject hypotheses based on term-specific thresholds instead of a single global threshold (Miller *et al.* 2007). These term-specific thresholds are set in a manner which maximises the actual term-weighted

value (ATWV) metric (NIST 2006), defined as follows:

$$\text{ATWV} = 1 - \frac{1}{|\mathcal{K}|} \sum_{K \in \mathcal{K}} \left(P_{\text{miss}}(K) + \beta P_{\text{fa}}(K) \right) \tag{5.1}$$

$$P_{\text{miss}}(K) = 1 - \frac{N_{\text{hits}}(K)}{N_{\text{true}}(K)}, P_{\text{fa}}(K) = \frac{N_{\text{fa}}(K)}{D - N_{\text{true}}(K)} \tag{5.2}$$

where $|\mathcal{K}|$ is the number of keyterms in the set $\mathcal{K}$, K is a particular keyterm, $N_{\text{hits}}(K)$ is the number of correct keyterm hypotheses (hits), $N_{\text{fa}}(K)$ is the number of incorrect hypotheses, $N_{\text{true}}(K)$ is the number of instances of the keyterm in the reference, D is the duration of the audio in seconds, and β is a parameter set to roughly 1000 (Vergyri *et al.* 2006). When considering the ATWV metric defined in equation 5.1, the benefit of accepting a hit is $\frac{1}{N_{\text{true}}(K)}$ and the cost of accepting a false alarm is $\frac{\beta}{|\mathcal{K}| - N_{\text{true}}(K)}$. Given a confidence score $c(d)$ for a keyterm hypothesis (detection) d, the expected benefit is defined as follows (Wang *et al.* 2012):

$$\xi(d) = \frac{c(d)}{N_{\text{true}}(K)} - \beta \frac{1 - c(d)}{|\mathcal{K}| - N_{\text{true}}(K)}. \tag{5.3}$$

Following the description in Wang *et al.* (2012), keyterm hypotheses with a positive value of $\xi(d)$ should be accepted to optimise the ATWV metric. The decision of whether to accept or reject a detection is then taken based on whether the value of the confidence score $c(d)$ is above or below the following optimal threshold:

$$\theta_K = \frac{\beta N_{\text{true}}(K)}{(\beta - 1) N_{\text{true}}(K) + |\mathcal{K}|} \tag{5.4}$$

where θ_K is estimated for, and applied to each keyterm K individually.

Instead of applying a term-specific threshold to the confidence scores, an alternative approach is to normalise the scores across all keyterms by mapping them in such a way that compensates for the differences between the score distributions for each keyterm. One technique which achieves this is that of rank-based normalisation across keyterms (Zhang *et al.* 2012). In this approach, the probability of false alarm (P_{fa}) is calculated for each keyterm, based on the output obtained when the system is applied to development data. The P_{fa} value for each keyterm is calculated by sorting the list of keyterms detections, and dividing the rank of the detection in the list by the total number of words in the data. A look-up table mapping confidence scores to P_{fa} values is built from this data. In test, the expected P_{fa} is obtained by querying this table, and the normalised score is taken to be $1 - P_{fa}$. The scores are

therefore normalised across keyterms as a function of their rank. In this way, scores are boosted for keywords which are biased towards underestimating and scores are decreased when the bias is towards overestimating.

The scores may also be scaled directly without the need to estimate P_{fa} on development data (Mamou *et al.* 2007). In this approach, a boosting vector is constructed such that the posteriors are scaled as a function of their rank in the list of detections (sorted by score). Similar to the approach taken in Zhang *et al.* (2012), elements of the boosting vector are chosen to be equal to the ratio $\frac{1}{i}$, where i is the depth of a keyword detection in the sorted list. Another approach which has been shown to be surprisingly effective in terms of the ATWV metric (Mangu *et al.* 2013), is to apply a sum-to-one normalisation to the scores. In this approach, the scores for each detection of a particular keyterm are normalised by the sum of the scores across all detections of the same keyterm.

5.2.3 Discriminative Score Mapping

As described in Section 5.2.2, there is a need for confidence scores associated with keyterm hypotheses to undergo some form of mapping to improve their accuracy. It is therefore natural to assume that this mapping can be learned effectively from data using discriminative techniques. There is a great deal of existing literature in the field of confidence estimation which addresses the problem of obtaining improved estimates of confidence measures for ASR hypotheses, some of which also makes use of discriminative models (see Chapter 3). This section will focus on the body of work in which these techniques have been applied to STD.

The primary reason for investigating techniques which apply discriminative score mapping, is that they are capable of reducing the effects of term dependence on the keyterm hypothesis score. For a start, training a model based on only the original score goes some way towards normalising confidence scores. This is due to the fact that a global mapping is learned which adjusts the scores such that they are normalised to be within similar ranges. Including keyterm-specific information in training the discriminative mapping can lead to further improvements, as this may be leveraged to further inform the mapping function as to the specific characteristics of individual keyterms. In Wang *et al.* (2009), multi-layer perceptrons (MLPs) and support vector machines (SVMs) were applied to this task. A mapping based on the original confidence score as the only feature, as well as one which includes

keyterm-specific features were investigated. For a specific keyterm k, the additional features used are the effective occurrence rate $R_0(K)$ and the effective false alarm rate $R_1(K)$, defined as follows:

$$R_0(K) = \frac{\sum_i c(d_i^K)}{T}, R_1(K) = \frac{1 - \sum_i c(d_i^K)}{T} \tag{5.5}$$

where d_i^K is the i-th detection d for keyterm K, and T is the length of the audio. This work showed that the MLP-based mapping slightly outperformed the SVM approach. Furthermore, some gains were seen when using $R_0(K)$ and $R_1(K)$ in combination with the confidence score c_u. In more recent work (Tejedor *et al.* 2010), the aforementioned approach is extended to incorporate additional features. The set of features used included lattice-based features (posterior score, R_0 and R_1), lexical information, edit distances, duration and prosodic features, as well as position features. The duration features include the duration of a keyterm hypothesis, the phone speech rate and the vowel speech rate. These features proved the most informative when combined with the lattice-based features.

An approach more akin to traditional ASR system combination may be used to improve the keyterm confidence scores of a system (Motlicek *et al.* 2012). In this work, the combination of the posterior scores output by different STD/KWS systems is proposed. This is achieved by aligning the hypothesised keyterms and their scores, and combining the scores using a discriminative mapping (maximum entropy, neural net, linear regression) to improved keyterm confidences. The linear regression approach was shown to yield the largest gains in this work.

Part II

Contributions

Chapter 6

Conditional Random Fields for Confidence Estimation

In this work, confidence estimation is approached as a binary classification task, in which the output from an underlying ASR system is effectively classified as either being correct or incorrect. A per-word score (confidence measure) indicating the degree to which each word in the output of the system is believed to be correct is sought. The use of a statistical classifier for this task allows for a well-defined probabilistic definition of the confidence measure. This measure is defined as the probability such a classifier would assign to the output of the ASR system being correct. A collection of features which are indicative of the reliability of the hypothesised transcription are used as the basis for making this classification.

In many LVCSR applications, the speech audio is segmented into sentences or *utterances* which are recognised as individual units. The transcription generated by the recogniser for each such utterance is therefore a sequence of words. In order to model and exploit the inherent sequential nature of the problem, a sequential classifier was chosen for use in this work. A contribution of this thesis is the application of a flexible conditional random field (CRF) framework to the CE task. These models are highly flexible, and possess a number of useful characteristics which further support their application to this task.

A central concept in this work is that multiple information sources (and the corresponding predictor features[1] derived from them), should be taken into account when estimating confidence scores. These predictor features are expected to be highly correlated. Generative models such as hidden Markov models (HMMs) are not suited to inputs which are highly correlated. Discriminative models on the other hand, such as conditional random fields (CRFs), are capable of dealing with features which are potentially highly correlated. This is due to their conditional nature. These models are therefore particularly amenable to situations in which multiple (potentially correlated) features are to be combined. CRF models are therefore well-suited for the approach taken in this work.

Another advantageous characteristic of CRF models that is of relevance for this work, is that arbitrary feature functions can be incorporated into the model. Feature functions can therefore be engineered that capture specific characteristics of the features and the data. These feature functions effectively allow the model to capture specific properties of the features accurately, and leverage them to further improve the ability of the model to represent the data accurately.

The sections which follow introduce some of the fundamental components of the flexible CRF-based framework for confidence estimation developed as a contribution of this work.

6.1 CRF Input: Predictor Features

In the CRF-based approach to confidence estimation of this work, information is required which can be used to inform the confidence estimation model, such that it can reason about the degree of confidence that should be assigned to a word. This information is encoded by so-called *predictor features*, which are in some way indicative of the likelihood of an ASR hypothesis being correct. These predictor features ultimately serve as the input to the CRF-based confidence estimation framework detailed in this work. Informative predictor features are therefore sought.

One obvious source of predictor features is an ASR system itself, which may be the system which produced the hypotheses that are to be annotated with confidence scores. Modern LVCSR systems combine information pertaining to the audio signal (represented by an acoustic model), with information on word sequences (represented by a language model) in order to transcribe (recognise) audio

[1] Features related to ASR hypotheses which are indicative of their reliability or correctness, and are thus useful in determining whether such hypotheses are correct will be referred to as predictor features.

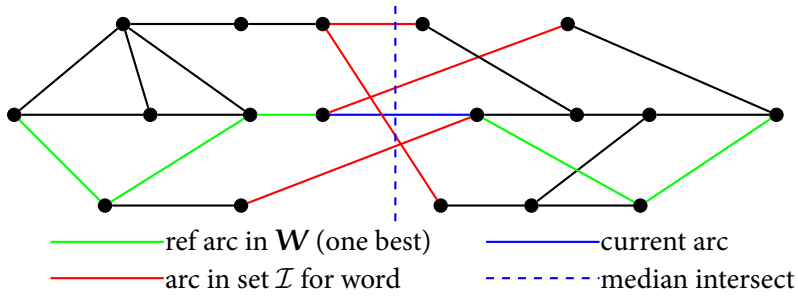


Figure 6.1 *Illustration of the process for extracting a set of relevant arcs for a word in a* 1*-best transcription (*W*), over which to compute predictor features.*

containing speech. These systems may also store detailed information pertaining to this recognition process, and the space of alternative hypotheses considered by the recogniser at each time within the audio, along with the relevant acoustic and language model scores. This information is stored in the form of recognition lattices (see Section 2.6 for a description). These lattices are therefore a rich source of information which can be processed to generate useful predictor features. A number of predictor features investigated in this work are obtained in this manner. In the sections which follow, a general procedure is presented for extracting such features from the single best transcription output by an ASR system, as well as for arbitrary alternative hypotheses.

6.1.1 *Feature Extraction from Lattices for 1-Best Hypotheses*

Confidence scores are associated with a specific hypothesis produced by an ASR system. The sentence-level maximum a-posteriori (MAP) criterion is often used during decoding with the Viterbi algorithm applied to ASR lattices in order to select the "best" hypothesis. A hypothesis generated from an ASR lattice corresponding to some audio in this way is referred to as the 1-Best hypothesis. Given a sequence of words in the 1-Best hypothesis, which essentially corresponds to a complete, connected path through the lattice, a general procedure for extracting predictor features from the lattices for each constituent word is proposed. This procedure is illustrated in Figure 6.1 and detailed in the discussion which follows.

1. A forward-backward pass is carried out on the recognition lattice. During this process posterior scores for all arcs in the lattice are computed. The forward pass also yields the 1-best hypothesis

($\boldsymbol{W}$) - which is shown in green in Figure 6.1 on the previous page.

2. For a given word at position i in the 1-best sequence $\boldsymbol{W}$, the specific arc in the path through the lattice which corresponds to that word in the 1-Best transcription is isolated (based on the timing information). This will be referred to as the current anchor arc for (W_i), and is shaded blue in Figure 6.1 on the preceding page.

3. All arcs in the lattice which intersect with the current anchor arc for W_i are found. An arc is considered to intersect with the anchor arc if it starts before and ends after the median time of this arc. This selection process is illustrated in Figure 6.1 on the previous page, where all arcs which intersect with the median intersect line (blue dashed line in the figure) are considered. The resulting intersecting arcs constitute the set $\mathcal{I}$, which are shaded red in the figure.

4. Given the set of arcs $\mathcal{I}$, which consists of all words hypothesised at approximately the same time as the word in the 1-best transcription under consideration (a local hypothesis space), predictor features are calculated.

5. This process is repeated for each word in the 1-best path through the recognition lattice.

The general procedure for computing a predictor feature for a hypothesised word W_i, given the set of arcs $\mathcal{I}$ which intersect with that word in the lattice is defined as follows:

$$\text{FEATURE}(W_i, \mathcal{I}) = \frac{\sum_{a \in \mathcal{I}} \delta(\text{word}(a), W_i) f(\cdot)}{\sum_{a \in \mathcal{I}} f(\cdot)} \tag{6.1}$$

where a feature is computed by applying some function $f(\cdot)$ to all arcs a in $\mathcal{I}$ with the same word identity word(a) as the reference word W_i (the match is imposed through the use of the Kronecker delta function δ). This sum is normalised in the denominator by the sum of the same function applied to all arcs in the set $\mathcal{I}$. A simple representative example of such a function is the identity function (i.e. $f(\cdot) = \mathbb{1}$). If this function is used, the resulting predictor feature will simply be the ratio of the number of matching word arcs to the total number of arcs in the intersecting set $\mathcal{I}$. The lattice-based predictor features investigated in this work will be expressed through the appropriate re-definition of the generic function $f(\cdot)$ to yield more specific forms of equation 6.1.

6.1.2 *Feature Extraction from Lattices for Alternative Hypotheses*

It is not always the case that the 1-best transcriptions generated from the lattices are those which yield the lowest word error rate. In fact, it is often the case that the hypotheses generated by carrying out techniques such as confusion network clustering result in improved hypotheses. However, this does not mean that the information available in the recognition lattice is no longer useful. A novel technique referred to as hypothesis injection is therefore proposed in this work, to extract features for such hypotheses from lattices.

In this technique, a supplied alternative hypothesis is essentially "projected" onto a recognition lattice to yield a simulated hypothesis. The procedure for obtaining this hypothesis is the following:

1. For a word hypothesis S_i at position i in the supplied alternative hypothesis $\mathbf{S}$, an anchor arc is sought in the lattice which has the same word identity and time registration information as S_i.

2. If such an anchor arc does not exist, a window is applied to the start and end times of S_i, which is increased in steps until a stopping criterion which specifies the maximum window size is met.

3. If the stopping criterion is reached, the word identity criterion is relaxed, and the arc which best matches the timing information is taken as the candidate anchor arc. For this arc, the posterior probability of the word in the supplied hypothesis will be 0, but it may still be informative to have other predictor features associated with this time span in the lattice.

The simulated hypothesis generated in this way consists of a sequence of equivalent word-level hypotheses from the lattice for a supplied alternative hypothesis, with similar time information. One significant distinction of the simulated hypothesis from the 1-best, is that it is not constrained to correspond to a path of connected arcs through the lattice, as such a path would not always exist. A further consequence of the projection approach is that some arcs may in fact overlap in time, but this is not problematic as the feature extraction approach treats each set of arcs for a given word-level hypothesis separately.

Once a simulated hypothesis is generated, a procedure similar to that carried out in the 1-best case is followed to extract predictor features. An illustration of this is shown in Figure 6.2 on the following page.

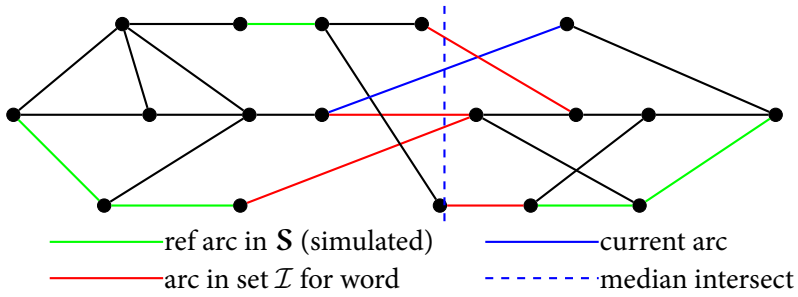


Figure 6.2 *Illustration of the process for extracting a set of relevant arcs for a word in a simulate transcription (**S**), over which to compute predictor features.*

6.2 CRF Architecture: Core Feature Functions

The general form of the CRF model was detailed in Section 4.2. A number of contributions are made in this work through engineering novel CRF feature functions. These feature functions are essentially specialised versions of the generic observation feature function which is expressed as follows:

$$g_{y',f(\cdot)}(\mathbf{Y},\mathbf{X},i) = \delta(\mathbf{y}_i, y')f(\mathbf{Y},\mathbf{X},i) \tag{6.2}$$

where y' is a "template" label for which the feature function is defined, $f(\cdot)$ is an arbitrary function applied to the input feature matrix $\mathbf{X}$, the label sequence $\mathbf{Y}$ and the observation index i within a sequence. Specific characteristics of the data can be captured by engineering new feature functions which define a suitably modified version of the function $f(\cdot)$. In this section, the core feature functions which are used in the CRF models developed in this work are described.

6.2.1 *Discrete Feature Representation*

CRF models were originally developed for use in natural language processing applications, such as in part-of-speech tagging. For such tasks, the features available for use are usually text-based, or are based on discrete attributes of the text. As a result, an observation feature function type which has a natural representation within the original CRF modelling framework is that of literal or discrete feature functions. For every possible value a literal predictor feature in the input can take, one such feature function template is instantiated per output label. These feature functions therefore match the value of the literal predictor feature for a particular observation i to the template value for each feature function.

The resulting feature function prototype is therefore defined as follows:

$$g_{y',l,d}(\mathbf{Y}, \mathbf{X}, i) = \delta(\mathbf{Y}_i, y')\delta(l, \mathbf{X}[d]_i) \tag{6.3}$$

where l is the particular literal value for which the feature function is defined, d is the dimension of the feature vector corresponding to the literal predictor feature and $\mathbf{X}[d]_i$ is therefore the value of this feature at position i in the observation sequence.

6.2.2 *Continuous Feature Representation*

The majority of predictor features investigated in this work are continuous in nature. Naturally, a discrete representation such as that applied in the case of the literal feature functions is not suitable as a representation of such predictor features. First order moment feature functions are one potential method for representing such data. These feature functions return the value of the continuous predictor feature, and are defined for each of the output labels. The prototype for these feature functions is defined as follows:

$$g_{y',d}(\mathbf{Y}, \mathbf{X}, i) = \delta(\mathbf{Y}_i, y')\mathbf{X}[d]_i \tag{6.4}$$

where d is the dimension of the feature function corresponding to a continuous predictor feature and $\mathbf{X}[d]_i$ is therefore the value of this feature at position i in the observation sequence. These feature functions are capable of representing a continuous predictor feature to some degree. However, they do not necessarily yield the best representation of such data, as they have limited expressive power within the standard maximum entropy framework. This is evident when one considers that the fundamental task of designing a system such that $E(f_k) = \tilde{E}(f_k)$ (a maximum entropy moment constraint matching the model and empirical expectancies), 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 not a strong enough constraint for system design, as was illustrated in Yu *et al.* (2009). One way this deficiency may be addressed to some extent by making use of second order statistics. The required feature function prototype for the second order moment is simply:

$$g_{y',d}(\mathbf{Y}, \mathbf{X}, i) = \delta(\mathbf{Y}_i, y')\mathbf{X}[d]_i^2. \tag{6.5}$$

The use of these second order moment functions makes it possible for the variance of a particular continuous feature to be modelled, but this may still not be as expressive as desired. The techniques outlined in the sections which follow attempt to resolve this deficiency by modelling the distribution more effectively with more parameters.

6.2.2.1 Moment Constraints with Binning

One 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 a sensible way. A technique described in White *et al.* (2007) 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 function $f(k, d, \mathbf{X}_i)$ defined on the interval $[l_{d,k}, h_{d,k}]$, is defined for each of the K quantisation intervals to yield binary functions of the following form:

$$f(k, d, \mathbf{X}_i) = \begin{cases} \frac{h_{d,k}+l_{d,k}}{2} & \text{if } \mathbf{X}_i^d \in [l_{i,k}, h_{i,k}] \\ 0 & \text{otherwise} \end{cases}, \tag{6.6}$$

where k is the bin index for which the function is defined, d is the dimension of the continuous feature in the vector at position i in the sequence of feature vectors $\mathbf{X}$. This function essentially yields a piecewise constant approximation to the continuous feature distribution, and is incorporated into the generic observation feature function framework to yield the following feature function prototype:

$$g_{y',k,d}(\mathbf{Y}, \mathbf{X}, i) = \delta(\mathbf{Y}_i, y')f(k, d, \mathbf{X}_i). \tag{6.7}$$

The parameters for these feature functions can be estimated in a standard way using the moment constraint matching approach.

A modification of this form of quantised bin features investigated in this work is one in which the actual value of the continuous feature is returned. This results in a hybrid between the first-order moment the quantised bin feature functions. Such feature functions effectively capture information pertaining to the true mean of the continuous feature within each bin, rather than assuming that the mean will be the value mid-way in the bin interval. The resulting quantisation function may therefore

be defined as :

$$f(k, d, \mathbf{X}_i) = \begin{cases} \mathbf{X}_i^d & \text{if } \mathbf{X}_i^d \in [l_{i,k}, h_{i,k}] \\ 0 & \text{otherwise} \end{cases}, \tag{6.8}$$

The form of the feature function prototype is the same as equation 6.7, using the redefined quantisation in 6.8.

Quantised bin features of this form have been implemented as part of the CRFTK software developed as part of this work (discussed in Section 6.4). In the implementation, the data is pre-processed such that all datapoints 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 White *et al.* (2007)). 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.

6.2.2.2 *Distribution Constraints with Cubic Splines*

Assuming that an infinite number of training samples are available, the number of bins K (as defined in Section 6.2.2.1) 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:

$$\lim_{K \to \infty} \sum_k \lambda_{ik} f_{ik} = \lambda_i(f_i) f_i \tag{6.9}$$

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 Yu *et al.* (2009), where it was proposed for maximum entropy models. With K knot points in the spline, a parametrised cubic spline approximation a_k may be derived for each knot k. As detailed in Yu *et al.* (2009), the product $\lambda_i(f_i)f_i$ may then be approximated as a sum of the products of K transformed features of the form $a_k(f_i)f_i$, weighted by K single-valued weights λ_{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

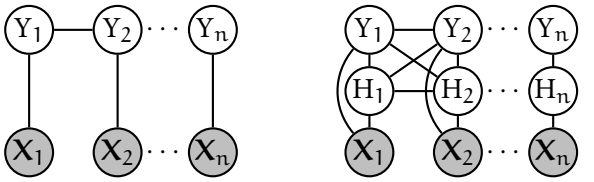
and the location of the knots, and is independent of the weights to be estimated. Such feature functions result in maximum entropy constraints for which the distributions are matched, and are therefore known as distribution constraints (DCs). Continuous feature functions defined in this manner have been shown to improve modelling accuracy over the binning technique detailed in Section 6.2.2.1 in a number of tasks Yu *et al.* (2009).

Spline feature functions with distribution constraints were also implemented as part of the CRF toolkit (`CRFTK`) developed in this work. The number of knots in the spline may be specified as a parameter for the feature instantiation process. The manner in which the spline knot points are placed may also be specified. The knot points may be placed based on a uniform segmentation of the interval, or on a policy which enforces uniform occupancy for data points within each interval.

6.3 Hidden-state CRF

For many tasks, a hidden structure is assumed to exist which should be modelled or accounted for. In such cases, statistical models which include hidden (latent) variables are useful. A hidden state model which is often employed, particularly in speech recognition, is the hidden Markov model (HMM). Similarly, hidden variables may be incorporated into CRF models. An overview of these models was presented in Section 4.3.2. One major benefit of CRF models with hidden states is that they do not impose strong independence assumptions on the input features, which is the case with HMMs. Furthermore, such models are still capable of supporting the use of arbitrary feature functions. As a result, CRF models with hidden states have been proposed for application in many tasks, such as phone classification (Gunawardana *et al.* 2005), phone recognition (Sung and Jurafsky 2009; Hifny and Renals 2009) and gesture recognition (Morency *et al.* 2007) to name a few. The use of such models for these tasks has generally shown to yield impressive performance gains. The CRF toolkit (`CRFTK`) developed in this work was extended so as to include hidden states within the model architecture. The application of these hidden-state CRF models to confidence estimation is a novel concept, and is therefore one of the contributions of this thesis.

Hidden states have been incorporated within general CRF models in a number of different ways, and have consequently been to referred to under various different names in the literature. The hidden



(a) Linear-chain CRF (b) CRF with Hidden states

Figure 6.3 *Graphical structure of the CRF and hidden-state CRF models used, where shaded vertices correspond to variables observed during training and test. The labels* $y_1 \dots y_n$ *are not observed in test (but are known in training), and are therefore unshaded. The hidden states* $h_1 \dots h_n$ *are not observed during training or test and are also unshaded.*

CRF (HCRF) models employed in Gunawardana *et al.* (2005) represent the joint distribution of a single "atomic" label y and a hidden sequence $\mathbf{h}$. This implies there is no transition structure at the output label level, and the model is applied to sequences of observations in which only one label is expected in the output. The form of hidden state CRFs proposed in this work model the joint distribution of an entire label sequence $\mathbf{Y}$ and hidden state sequence $\mathbf{h}$, all conditioned on the sequence of observation vectors $\mathbf{X}$. Thereby addressing the task of assigning a sequence of labels, such that the output label transition structure is captured as well as the hidden-state transition structure. The resulting model architecture is shown in Figure 6.3b.

For the CE task, $\mathbf{X}_i$ represents a sub-word-level or word-level vector of predictor features (at position i), Y_i represents the corresponding discrete label (`Correct` or `Incorrect`) and H_i is the hidden variable related to $\mathbf{X}_i$ and Y_i. Marginalising out over hidden state sequences $\mathbf{H}$ in the set of possible sequences $\mathcal{H}$ yields the following:

$$p(\mathbf{Y}|\mathbf{X}) \propto \sum_{\mathbf{H}\in\mathcal{H}} \exp\Big(\sum_k \lambda_k t_k(\mathbf{Y},\mathbf{H}) + \sum_l \mu_l g_l(\mathbf{Y},\mathbf{H},\mathbf{X})\Big)$$

where $t_k(\mathbf{Y},\mathbf{H})$ are the transition feature functions and $g_l(\mathbf{Y},\mathbf{H},\mathbf{X})$ are the observation feature functions, with parameters λ_k and μ_l respectively. Each of the k transition feature functions corresponds to a template combination of previous/current label and hidden state values (y', y, h', h). Each of the g_l feature functions takes the current observation vector $\mathbf{x}_i$ as an argument, and corresponds to a template current label and hidden state value (y, h). These are enumerated over the set of discrete values a predictor feature can take, or a quantisation parameter in the continuous case. The feature functions

are expressed as follows:

$$t_{y',y,h',h} = \delta(Y_{i-1}, y')\delta(H_{i-1}, h')\delta(Y_i, y)\delta(H_i, h) \tag{6.10}$$

$$g_{y,h}(\mathbf{x}_i) = f(\mathbf{x}_i)\delta(Y_i, y)\delta(H_i, h) \tag{6.11}$$

where i is an index within the observation/label sequence, δ is the Kronecker delta function, and $f(\mathbf{x}_i)$ is a function applied to the current observation vector $\mathbf{x}_i$. This function is dependent on the specific type of observation feature function. The model parameters λ_k and μ_l are estimated by optimising the conditional log-likelihood of the model using a gradient-based technique (limited-memory BFGS (Liu and Nocedal 1989)).

Although the structure of the model described is similar to that described in Morency *et al.* (2007), the values the hidden variables could take were constrained in the latter work to be in a disjoint set for each label. This implies that a particular value of the hidden state variable is constrained to only occur with a particular value of the label variable. This restriction is optionally relaxed in the architecture proposed here, which results in a more general model.

In this thesis, the notion that regions of confidence tend to exist in ASR output is put forth. This approximate or "fuzzy" segmentation of words into sequences of consecutive correct or incorrect words can be thought of as some hidden structure. As a result, the hidden-state CRF model described in this section is applied to the confidence estimation task, such that the modelling power proffered by the hidden state space can be used to capture the desired latent dynamics of error sequences.

6.4 CRF Implementation

Many of the useful characteristics of CRF models are difficult or even impossible to exploit using standard, off-the-shelf implementations of CRFs. One deficiency in most implementations is in the approach taken to deal with continuous predictor features. Adequate mechanisms for representing such inputs are often not present, or are otherwise not sufficiently expressive. Furthermore, the ability to engineer arbitrary feature functions is severely limited. A contribution of this work is therefore the development of a highly flexible bespoke CRF toolkit in the C# language (`CRFTK`). Other than the feature engineering aspect, the development of this toolkit also made it possible for CRF models with more complex,

interesting architectures to be developed and experimented with. Examples of such extensions include support for CRFs with hidden states and skip-edge feature functions.

The CRF toolkit was developed in such a manner as to be as flexible and powerful as possible, such that the experimentation required for this thesis could be carried out effectively. Some of the more interesting characteristics of the model in a general functional sense are summarised below.

- The feature vector of observations associated with each observation may consist of features with different base types. In other words, the feature vectors may consist of an arbitrary mix of continuous and discrete features. This is managed by specifying the variable type which should be used to represent each dimension of the observation vector in the input data.

- Any combination of the set of possible feature functions supported by the toolkit may be applied to any suitable dimensions of the feature vector. The feature function combinations are defined when instantiating a model structure for training.

- Sparse vector representations and matrix algebra are used in computing the necessary statistics during training, which greatly reduces the space and time requirements of the model training process.

- Parameter estimation (training) may be carried out in parallel over an arbitrary number of different threads or processes, to further expedite the training process. The parallelisation is carried out over the data, with each process computing statistics for a different subset of the data.

- Parameters of the training process such as those governing convergence criteria and the form of regularisation used may be specified.

6.5 Evaluation

6.5.1 Word and Utterance-level Metrics

A metric which is often used to evaluate the performance of confidence estimation systems is the normalised cross entropy (NCE) (Siu *et al.* 1997), which was initially introduced in Section 3.4. This metric

evaluates the quality of the confidence score distribution on the word level. The NCE score is a measure of the information gain achieved when assigning the confidence score obtained from the system under evaluation to each word, rather than the empirical error rate of the underlying ASR system (which is a naive assumption of confidence). The NCE metric is defined as follows:

$$\mathrm{NCE} = \frac{\left(H_{\max} + \sum_{w \in \mathcal{C}} \log_2(\hat{P}(w)) + \sum_{w \in \mathcal{I}} \log_2(1 - \hat{P}(w))\right)}{H_{\max}} \tag{6.12}$$

where $\hat{P}(w)$ is the confidence score for a particular word w, $\mathcal{C}$ is the set of correct words and $\mathcal{I}$ is the set of all incorrect words. Given n correct words in the hypotheses out of N total words the empirical average probability of a word being correct is $P_c = \frac{n}{N}$, and

$$H_{\max} = -n \log_2(P_c) - (N - n) \log_2(1 - P_c). \tag{6.13}$$

However, the NCE metric is defined based solely on individual words, and does therefore not provide insight into the performance of confidence measures over word sequences. There are applications in which it is potentially advantageous to have some sense of the quality of confidence scores assigned to sequences of words, or sub-sequences (phrases). One such example is in a semi-automatic transcription service, where more accurate scores on this level would lead to improved decisions on which segments or utterances to forward to manual transcribers when the confidence level is low. This observation prompted the development and use of a measure relating the average confidence score of an utterance with the empirical error rate of that utterance in this work. Given a dataset of S utterances, in which each utterance is of length L_i, the utterance-level mean absolute deviation (UMAD) (Seigel and Woodland 2011) may be defined as:

$$\mathrm{UMAD} = \frac{1}{S} \sum_{i=1}^{S} \left[\frac{1}{L_i} \left| \sum_{j=1}^{L_i} c_{ij} - \sum_{j=1}^{L_i} \alpha_{ij} \right| \right]$$

where for word j in utterance i, c_{ij} is the ideal confidence value (0 for incorrect words or 1 for correct words), and α_{ij} is the corresponding confidence score under evaluation.

6.5.2 *Measuring Significance*

The metrics described in Section 6.5.1 are useful in assessing the performance of confidence estimation systems. This is particularly relevant when determining the impact of changes made in such systems,

and when determining the relative rankings of these systems. In this work, contrasts are frequently drawn between systems developed to improve the quality of confidence scores based on these metrics. It is therefore of interest to determine whether or not the performance of two systems is expected to be different when applied to different data, thereby implying the results obtained are indeed statistically significant. Whilst it is certainly possible to draw conclusions based on the relative rankings of CE systems using various metrics, a more rigorous analysis of the results is carried out in this work to yield further insights. The problem of statistical significance testing for confidence estimation system results has not been addressed in the existing literature. The formulation of such a significance testing procedure is thus a contribution of this thesis.

A core principle of any statistical significance testing technique, is that of the null hypothesis (H_0). The aim of the significance test is to determine the probability or confidence with which this null hypothesis may be rejected. In this work, the null hypothesis is based on the NCE score of two systems under consideration. This hypothesis may be stated as being that the mean of the differences between the NCE scores of the two systems is zero. If this null hypothesis is proven to be true, it implies that the two systems are effectively the same (or at least so similar that their difference is not significant). This test is developed following on from the definition of the matched pair significance test commonly employed for significance testing in terms of WER metrics for ASR output (Gillick and Cox 1989). It is assumed that there is some segmentation of the ASR output which produces K segments or regions for which the NCE scores are statistically independent of each other. Given two confidence estimation systems (A and B), the NCE scores for a particular segment i for each system are:

$$\mathrm{NCE}_A^i = \frac{\left(H_{max} + \sum_{w \in \mathcal{C}_i} \log_2(\hat{P_A}(w)) + \sum_{w \in \mathcal{I}_i} \log_2(1 - \hat{P_A}(w))\right)}{H_{max}}$$

$$\mathrm{NCE}_B^i = \frac{\left(H_{max} + \sum_{w \in \mathcal{C}_i} \log_2(\hat{P_B}(w)) + \sum_{w \in \mathcal{I}_i} \log_2(1 - \hat{P_B}(w))\right)}{H_{max}}$$

where $\mathcal{C}_i$ and $\mathcal{I}_i$ are the sets of correct and incorrect words in segment i respectively, the scores assigned by system A and B to a word w are $\hat{P_A}(w)$ and $\hat{P_B}(w)$ respectively and the "empirical" entropy H_{max} is defined as :

$$H_{max} = -n_i \log_2(P_c) - (N_i - n_i) \log_2(1 - P_c) \tag{6.14}$$

where N_i is the total number of words in segment i, n_i is the number of correct words in the segment and $P_c = \frac{n}{N}$ is the empirical probability of a word being correct over all segments. The test statistic which is defined by the null hypothesis is the difference in the NCE values for the systems for a segment i:

$$Z_i = NCE_A^i - NCE_B^i$$

The mean difference in the NCE scores for each system is μ_Z, which is defined as:

$$\hat{\mu}_Z = \sum_{i=1}^{K} \frac{Z_i}{K} \tag{6.15}$$

and the variance estimate of the Z_i values is therefore:

$$\sigma_Z^2 = \frac{1}{K-1} \sum_{i=1}^{K} (Z_i - \hat{\mu}_Z)^2. \tag{6.16}$$

The test parameter W may then be defined as:

$$W = \frac{\hat{\mu}_Z}{\frac{\sigma_Z}{\sqrt{K}}}. \tag{6.17}$$

For large values of K, the distribution of W can be approximated by a zero-mean normal distribution with unit standard deviation. For a two-tailed significance test, the p-value is defined as follows:

$$p = 2p(X \geq |W|) \tag{6.18}$$

where X is a random variable which has the form of the standard normal distribution $\mathcal{N}(0, 1)$. The differences in the output of two systems may be considered statistically significant if this P value is less than a desired level of confidence α, which signifies that the null hypothesis H_0 of there being no difference between the systems in question may be rejected. Typical values of α are 0.05 (5% confidence level), 0.01 and 0.001.

In each of the system pairs evaluated for significance testing, one system will yield an improved (or equal) NCE score in relation to that of the other system. A measure of the degree to which one system yields improved performance over the other on a per-snippet basis is proposed. This snippet improvement percentage (SIP) is computed as the number of segments (snippets) for which the system of interest in the significance test yields an improved NCE score over the alternate system, divided by

the total number of segments. This ratio is therefore expressed as follows:

$$\mathrm{SIP}(A, B, \mathcal{S}) = \frac{\sum_{S \in \mathcal{S}} \delta(\mathrm{NCE}_A(\mathbf{S}) > \mathrm{NCE}_B(\mathbf{S}))}{\sum_{S \in \mathcal{S}} \mathbb{1}} \tag{6.19}$$

where $\mathbf{S}$ is a particular snippet in the entire set of snippets $\mathcal{S}$ within the audio. System A is assumed to be the system of interest in the test which achieves an NCE score of $\mathrm{NCE}_A(\mathbf{S})$ on snippet $\mathbf{S}$, and $\mathrm{NCE}_B(\mathbf{S})$ is the NCE score achieved by the alternate system on this snippet. The Kronecker delta function is used to test the condition of whether the the NCE score for the system of interest ($\mathrm{NCE}_A(\mathbf{S})$) is larger than that of the alternate system ($\mathrm{NCE}_B(\mathbf{S})$), and returns 1 if this is true and 0 otherwise.

6.5.3 *Decision Tree Baseline*

A sophisticated baseline was sought as a point of comparison for the NCE results achieved by the CRF-based confidence estimation investigated in this thesis. The simplest baseline would be to use a single predictor feature extracted from the recognition lattices, such as the word posterior, as the confidence score. This is however not a particularly useful baseline, as these scores typically overestimate the true confidence level. Instead, the approach taken in Evermann and Woodland (2000*b*) was employed. In this approach, the posterior probabilities for word hypotheses are re-mapped to appropriate confidence scores. This is achieved by firstly training a decision tree classifier using the posterior feature as the input. The resulting decision tree, and in particular the set of splits or questions used to generate the leaf nodes, is used to define a set of quantisation intervals or "bins" along the real line, into which the posterior score in question should be quantised. Thereafter, a linear mapping is applied over each of these intervals. The resulting piecewise-linear mapping of confidence scores was shown to perform well, and is the form of the first baseline system against which contrasts are drawn in this thesis. The approach employed by the baseline system does not however scale to situations in which multiple predictor features are to be considered when estimating a confidence measure, as the piecewise linear mapping cannot simply be achieved in multiple dimensions simultaneously. Furthermore, the baseline approach does not account for any sequential aspects of the task.

Chapter 7

Word-level Confidence Estimation

The confidence estimation task is often defined on the word level, where the aim is that of estimating confidence scores for individual words of a given ASR hypothesis. This is a result of the fact that LVCSR systems typically output word-level transcriptions for audio. These transcriptions are subsequently processed in this form by downstream applications, which may reap benefits from information on the level of confidence in each word.

In the context of the general approach taken in this work, a set of predictor features which are defined on the word level are therefore used as input to a CRF model. Based on this input, the CRF models have the task of assigning labels to each word in the hypothesis. For confidence estimation, these labels indicate whether or not the present word is classified as being correct or incorrect. The marginal probability of the label corresponding to the word being correct is computed, and is used as the confidence score for that word.

Word-level confidence estimation is explored in this chapter. The CRF-based framework is developed to address this task, and is described in detail. Thereafter, the results of a series of experiments carried out with the aim of investigating various aspects of the problem, and the overall utility of the approach, are presented and discussed.

The contributions presented in this chapter are based primarily on publications detailing the word-level framework and application of CRFs for confidence estimation (Seigel and Woodland 2011), and the hidden-state CRF model for this task (Seigel and Woodland 2012).

In work published while parts of the work presented in this chapter were ongoing, CRF models were applied to a word-level CE task (Fayolle *et al.* 2010). However, the approach taken in this work is significantly different in three main ways. Firstly, the focus in Fayolle *et al.* (2010) is on using recogniser-independent predictor features defined at a high-level of abstraction. In this work, both recogniser-dependent predictor features (extracted from lattices), and other recogniser-independent high-level features of various types are used. Secondly, only predictor features of a discrete nature were considered in the related work, with no provision being made for continuous predictor features. In this work, specialised feature functions are developed to represent the many informative continuous predictor features available. Both these representations and the information encoded by these continuous features are shown to be be crucial in yielding improved confidence scores. Thirdly, the CRF model is refined here through the development a number of specialised feature functions, as well as through the extension of the model to include hidden states, which yields a more expressive statistical model overall.

7.1 Predictor Features

Predictor features are measures or features that are somehow related to the likelihood of a particular word hypothesised by an ASR system truly being correct (i.e. having been present in the original audio at the same time as the hypothesis). Such predictor features therefore serve to inform the confidence estimation process. In this section, a number of word-level predictor features investigated in this work are detailed. It should be noted that this is not an exhaustive study of all possible predictor features that may be used for confidence estimation. Instead, the predictor features that are discussed and utilised represent a selection of those which are either commonly used, or are particularly interesting and useful within the context of this thesis. Some of these are novel predictor features introduced as part of the contributions in this work.

7.1.1 Word-level Predictor Features from Lattices

A number of the predictor features introduced here are extracted from word-level recognition lattices. The general procedure for their calculation is based on the arc intersect method also introduced in

this work, which is detailed in Sections 6.1.1 and 6.1.2. In order to compute a particular feature of interest, some operation is carried out over the arcs which intersect with a given word in the lattice. This intersecting set of arcs may be thought of conceptually as a "local hypothesis space". More specifically, predictor features are generally computed in relation to such a "reference" hypothesis word[1] W_i, at the i^{th} position in a sequence of words $\boldsymbol{W}$ hypothesised by the recogniser. The general formula for a predictor feature related to this reference word W_i was first defined in equation 6.1 as:

$$\text{FEATURE}(W_i, \mathcal{I}) = \frac{\sum_{a \in \mathcal{I}} \delta(\text{word}(a), W_i) f(\cdot)}{\sum_{a \in \mathcal{I}} f(\cdot)} \tag{7.1}$$

where $\mathcal{I}$ is the intersecting set of lattice arcs, $\text{word}(a)$ is the word identity of the arc a, and $f(\cdot)$ is the key function defining the measure which is ultimately calculated. In the sections which follow, predictor features of interest will be introduced, and expressed through the appropriate definition of specialised versions of the function $f(\cdot)$. These are generally a function of information corresponding to arcs in the lattice. Examples of this information include the acoustic and language model scores and the word identity.

7.1.1.1 Lattice-based Posterior Probabilities

The posterior probability of a given word having occurred at some time in the audio, based on the information provided by the acoustic and language models used by the ASR system, is naturally a good measure of confidence for that particular hypothesis being correct. Time-based posteriors may be computed based on the information in recognition lattices, as described in Evermann and Woodland (2000*b*). Following the derivation of these posteriors in that work, the joint probability of a path $\boldsymbol{q}$ through the recognition lattice corresponding to an utterance hypothesis $\boldsymbol{W}$ and an acoustic observation vector $\boldsymbol{X}$ is composed of the acoustic model likelihood $p_{ac}(\boldsymbol{X}|\boldsymbol{q})$, the language model likelihood $P_{lm}(\boldsymbol{W})$, and the pronunciation likelihood $P_{pr}(\boldsymbol{q}|\boldsymbol{W})$:

$$p(\boldsymbol{q}, \boldsymbol{X}) = p_{ac}(\boldsymbol{X}|\boldsymbol{q})^{\frac{1}{\gamma}} P_{lm}(\boldsymbol{W}) P_{pr}(\boldsymbol{q}|\boldsymbol{W}) \tag{7.2}$$

where γ is the grammar scaling factor which specifies the weighting between the acoustic and language model scores. The link (or lattice arc) posterior may then be defined as the sum of the probabilities

[1] This should not be confused with the reference against which ASR hypotheses are scored. Here, the reference is a word in the lattice for which predictor features are extracted.

over the set Q_a of all paths in the lattice $\mathbf{q}$ which include the arc a, normalised by the probability of the acoustic observation $p(\mathbf{X})$:

$$P(a|\mathbf{X}) = \frac{\sum_{q \in Q_a} p(\mathbf{q}, \mathbf{X})}{p(\mathbf{X})} \tag{7.3}$$

where $p(\mathbf{X})$ is estimated as the sum over all paths.

The time-based arc posteriors (in equation 7.3) represent the posterior of a particular word arc, and are calculated using the probabilities of all paths through the lattice which pass through that arc. However, there may be arcs with the same word identity as this single arc, which are hypothesised as having started or ended at some time which is only marginally different. The arcs corresponding to these additional hypotheses will thus lie on different paths through the recognition lattice, and are therefore not considered in the computation of time-based posteriors. These posterior probabilities are therefore accurate in the sense that they represent the posterior for a particular word hypothesis, defined on a strict time interval. For confidence estimation, the exact timing of the word hypothesis is not as important, and a more accurate predictor of "word correctness" is one which accounts for hypotheses of the same word which overlap in time. The lattice arc posterior ratio (LAPR) predictor feature introduced in this work accounts for these overlapping word hypotheses to yield accurate word posteriors. This is achieved by relaxing the time constraints slightly and re-normalising the time-based arc posteriors defined in equation 7.3. Here, the word-level arc posterior ratio for a word W_i in the reference hypothesis $\boldsymbol{W}$ is calculated by summing the posterior probabilities $p(a|\mathbf{X})$ for all arcs in the intersecting set $\mathcal{I}$ with the same word identity, word(a), as the hypothesised word W_i. This sum is normalised by the sum of the posterior probabilities for all arcs in $\mathcal{I}$ yielding:

$$\text{LAPR}(W_i, \mathcal{I}) = \frac{\sum_{a \in \mathcal{I}} \delta(\text{word}(a), W_i) p(a|\mathbf{X})}{\sum_{a \in \mathcal{I}} p(a|\mathbf{X})} \tag{7.4}$$

where δ is the Kronecker delta function, used here to match the word identities of the reference arc and a.

The language and acoustic models are assumed to work in a complementary fashion when assigning scores to hypotheses (and thereby defining the hypothesis space that is not pruned away). However, in some instances, the evidence for a particular hypothesis may be supported more by one model than the other. This leads to some level of disagreement between the models, such that the relative scores may be vastly different. In such cases, the word posterior score (LAPR) might indicate the hypothesis

is more likely than it should perhaps be. This is a result of the information in the difference between the model scores effectively being lost through the averaging that takes place when combining the scores from these models when computing the arc posterior. A predictor feature which serves to decouple the language model scores from the acoustic model scores is therefore proposed in this work. This feature, which is named the lattice arc language model posterior (LALMP), makes use of the language model scores P_{lm} in isolation and is defined as follows:

$$\text{LALMP}(W_i, \mathcal{I}) = \frac{\sum_{\mathcal{I}} \delta(\text{word}(a), W_i) P_{lm}(\boldsymbol{W})}{\sum_{\mathcal{I}} P_{lm}(\boldsymbol{W})}. \tag{7.5}$$

At any time during the recognition process, a number of different hypotheses are considered in parallel by a typical recogniser. When a large number of different words are hypothesised at the same time, this may signify a situation in which the recogniser is more uncertain in the single word hypothesis ultimately selected over this time span. The structure of the recognition lattice itself may be used to estimate a measure of the level of "confusion" experienced by the ASR system. This measure is based on the identity of the different words which constitute the set of competing lattice arcs $\mathcal{I}$ for a hypothesised word. Based on this information, the degree to which the recognition process is having to consider alternate paths consisting of different word hypotheses at a particular time may be quantified. This results in what shall be referred to as the word confusability ratio (WCR). The WCR predictor feature is computed over the same set of arcs for a given reference word in the lattice as the LAPR predictor feature. However, the raw (un-weighted) counts of word occurrences are used. This is different from LAPR where the posterior score is utilised for each occurrence, such that these occurrences are effectively weighted by their posterior. These predictor features would therefore be equivalent if the posteriors of all word arcs were assumed to be 1.0. The WCR predictor feature is a measure based on the raw number of competing hypotheses, and is not skewed by the probability mass corresponding to the 1-best word as LAPR might be. This predictor feature is therefore expressed as follows:

$$\text{WCR}(W_i, \mathcal{I}) = \frac{\sum_{a \in \mathcal{I}} \delta(\text{word}(a), W_i) \mathbb{1}}{\sum_{a \in \mathcal{I}} \mathbb{1}} \tag{7.6}$$

where $\mathbb{1}$ is the unit identity function, which always returns the value 1.

It is possible that the number of alternate hypotheses being considered by the recogniser at a given time are slightly different spellings or surface forms of very similar words. In such cases, it may be useful to know that the hypothesis space is dominated by a low number of similar words. A predictor

feature which captures this information by calculating the posterior probability of words similar to the reference word is therefore proposed. This posterior is computed by modifying the definition of the reference word against which the identity of word arcs is matched in the standard LAPR computation. In computing this predictor feature, a set of words which includes all words within a certain edit distance of the original word, are considered as being the hypothesised "reference word". The resulting similarity posterior (SP) for a particular maximum edit distance e is therefore defined as follows:

$$\mathrm{SP}(W_i, e, \mathcal{I}) = \frac{\sum_{a \in \mathcal{I}} \sum_{s \in \mathcal{E}(W_i, e)} \delta(\mathrm{word}(a), s) p(a|\mathbf{X})}{\sum_{a \in \mathcal{I}} p(a|\mathbf{X})} \tag{7.7}$$

where there is a sum over each word s in the set of words $\mathcal{E}(W_i, e)$, defined as those words in $\mathcal{I}$ within a maximum edit distance of e from the hypothesised word W_i.

In Section 8.1 it is put forward that an alternative recogniser, acting on a level other than the word level (i.e. the sub-word-level), may yield useful alternative information for confidence estimation. One particularly useful predictor feature is the alternative recogniser's sub-word posterior (ASWP) for the sub-word hypothesised in the 1-best path through the lattice, expressed in equation 8.2. Another predictor feature of this type is the best alternative sub-word posterior (BASWP), which is the posterior score for the highest-scoring sub-word unit hypothesised by the alternative recogniser, and is defined in equation 8.4. These predictor features are naturally useful in the context of sub-word-level confidence estimation, discussed in Chapter 8. The ASWP and BASWP predictor features are extracted from a source which is entirely separate from the underlying recogniser. They are therefore capable of contributing new, different information to the confidence estimation process, which may lead to improvements in the word-level task. Understanding the impact of these features for the word-level confidence estimation task is important, such that it can be contrasted with that in the sub-word-level task (see Section 8.4). In order to make use of these predictor features here, they must be converted to a word-level representation. This may be achieved by simply averaging the scores up over the constituent sub-words corresponding to a word. The resulting averaged posteriors AASWP and ABASWP may be expressed as follows:

$$\mathrm{AASWP}(W_i, \mathcal{I}) = \frac{\sum_{G \in W_i} ASWP(G, G_{t_b}, G_{t_e})}{\sum_{G \in W_i} \mathbb{1}} \tag{7.8}$$

$$\mathrm{ABASWP}(W_i, \mathcal{I}) = \frac{\sum_{G \in W_i} BASWP(G_{t_b}, G_{t_e})}{\sum_{G \in W_i} \mathbb{1}} \tag{7.9}$$

where the average is taken over each sub-word G comprising the word W_i and G_{t_b} and G_{t_e} are the beginning and end times of the sub-word unit G respectively.

7.1.1.2 Lattice Acoustic Stability

A given hypotheses may be supported equally by both the language and acoustic model components of the recogniser. If this is the case, the hypothesis may be considered to be "stable". For such hypotheses, if the weighting of the acoustic and language model were changed, this should not have a significant impact on the resulting hypothesis space represented in the lattice (and consequently the 1-best hypothesis). The stability of word hypotheses may be quantified using the acoustic stability metric introduced in Zeppenfeld and Waibel (1992). Here, the degree to which the hypothesis space changes when using a range of different values for the weighting of the language and acoustic models in the arc posterior calculation is measured. This weighting is known as the grammar scaling factor γ. The acoustic stability is calculated by generating a set of N alternative hypotheses, each of which is obtained by decoding with a different value for γ, to produce a 1-best hypothesis. These different hypotheses are then aligned with that resulting from the use of the original tuned γ value. The acoustic stability for each word is then calculated as the ratio of the number of matching aligned words to the total number of words, N.

A novel method for calculating a measure of acoustic stability "on the fly" which does not necessitate the generation and subsequent alignment of N different hypotheses is proposed in this work. Arc posterior scores are calculated separately for each of the N different γ values during a single forwards-backwards pass over the lattice, and are attached directly to the arcs in the lattice. The set of intersecting arcs $\mathcal{I}$ (representing the localised hypothesis space) for each hypothesised word W_i is sorted along the dimensions of the vector of posterior scores corresponding to each γ value. A percentage (taken to be 75% in this work) of the top ranked arcs corresponding to each γ value is then taken to form the subsets $\mathcal{T}_j$, each of size L_T. The word-level acoustic stability for each word W_i is calculated by summing the number of matching arcs in each subset $\mathcal{T}_j$, and normalising by the total number of arcs in each of the N subsets to yield the lattice acoustic stability (LAS):

$$\mathrm{LAS}(W_i, \mathcal{I}) = \frac{\sum_{j=1}^{N} \sum_{a \in \mathcal{T}_j} \delta(\mathrm{word}(a), W_i)}{N \times L_T}. \tag{7.10}$$

The degree to which the number of matched arcs in each subset changes is the quantity represented by this predictor feature. This effectively gives some indication of the "stability" of the local hypothesis space for a word in the lattice, in terms of how likely it is to change. For instance, in noisy conditions, the acoustic model scores may be expected to be less accurate. Changing the scaling factor will result in a large number of scores for hypotheses the acoustic model is unsure of being boosted or discounted, resulting in significant changes in the hypothesis space. This will result in a low degree of stability, which should indicate a region where an error is more likely to have occurred.

The lattices produced by ASR systems represent multiple different potential hypotheses at any one time. These competing hypotheses can be thought of as "confusions", and are a useful source of information. An extension of the acoustic stability concept is proposed to make use of this information. Here, a measure is made on the degree with which the posterior probability of the reference word deviates from that of the posterior probability of a competing word, evaluated using different scaling factors in the posterior calculation. This deviation provides some indication of whether the posterior probabilities of confusable words and the reference words tend to change in step with changes in the scaling factors, or whether they vary to a significant degree. Intuitively, a great deal of variation and a correspondingly large deviation, should imply that the reference hypothesis is indeed not stable, and therefore less likely to be correct.

In this work, the two top-scoring competing words or "confusions" relating to a specific reference word are considered. The LAPR feature is computed both for the reference word, and each of the 2 competing words (if they exist), using each of the grammar scaling factors. The lattice posterior stability (LPS) feature is therefore defined as the average deviation between the posterior score for the reference word and the competing word, evaluated for each grammar scaling factor. This predictor feature is expressed as follows:

$$\mathrm{LPS}(W_i, d, \mathcal{I}) = \frac{\sum_{j=1}^{N} |\mathrm{LAPR}_{\gamma_j}(W_i, \mathcal{I}) - \mathrm{LAPR}_{\gamma_j}(C_d, \mathcal{I})|}{N} \tag{7.11}$$

where C_d is the competing word at position d in the list of competitors, N is the number of grammar scaling factors, γ_j is one of the grammar scaling factors for which the LAPR posterior is calculated, and W_i is the reference word for which the LPS predictor feature is being computed.

A measure which quantifies the stability of the scores for a particular competing word itself, rather than how "stable" these are in relation to the reference word (as in the case of LPS) is expected to be

informative. An autocorrelation-like stability metric, which is similar to the LPS predictor feature is therefore proposed for this purpose. This metric is computed across the posteriors corresponding to different grammar scaling factors, over the set of scores related to a particular competing word. The resulting word stability (WS) predictor feature is therefore defined as follows:

$$\mathrm{WS}(d, \mathcal{I}) = \frac{\sum_{j=1}^{N} \sum_{k=1}^{N} |\mathrm{LAPR}_{\gamma_j}(C_d, \mathcal{I}) - \mathrm{LAPR}_{\gamma_k}(C_d, \mathcal{I})|}{N \times N} \tag{7.12}$$

where there are two sums over the N different grammar scaling factors.

7.1.2 *Term and Document Frequency*

The majority of the predictor features detailed thus far have focussed on information local to a given word hypothesis at a particular time. They have also been extracted from a single recognition lattice corresponding to the utterance within which the hypothesis occurs. However, long-range information can have a significant effect on the level of confidence that should be assumed in this hypothesis. For instance, it is likely that certain themes or topics will reoccur in successive portions of audio recordings, resulting in language that is repetitive (e.g. within a news report, or within a conversation). Typical ASR systems are unable to feasibly account for this long-term dependency aspect of language. This information can however be incorporated into the confidence estimation process, and thereby capitalised upon to improve confidence score estimates.

In the field of information retrieval (IR), the use of long-range information over collections of documents is commonplace, and has been used extensively. One particularly useful measure used in IR is known as the TF*IDF score (Spärck Jones 1972). This measure is often utilised in order to rank documents in terms of relevance for a particular search query. In this work, this score is proposed as a method through which some notion of long-range dependencies in language can be incorporated into the confidence estimation process.

In order to compute the necessary statistics required by TF*IDF, each individual show[2] "snippet", which is a subset of the utterances within the show, is considered to be a document. The TF*IDF score is subsequently computed for each word within the utterance hypotheses output by an ASR system, over each snippet in the audio. The term frequency, TF, for a given term T (a word) in a document D

[2] A show in this context is is a broadcast news show.

(which is a snippet within the audio data), may be defined as follows:

$$\mathrm{TF}(T, D) = f(T, D) = \sum_{W \in D} \delta(W, T) \tag{7.13}$$

where each word W in the document is matched using the Kronecker delta function δ against the term of interest T in order to compute the number of times it occurs in the document. The inverse document frequency (IDF), which is a measure of how common a term is to occur over all documents, is computed for a given term T and a collection of documents $\mathcal{D}$ (i.e. all snippets in the audio corpus) as follows:

$$\mathrm{IDF}(T, D) = \log \frac{|\mathcal{D}|}{|\{D \in \mathcal{D} : T \in D\}|}. \tag{7.14}$$

7.1.3 *Confusion Network Posteriors*

In confusion network (CN) clustering, a "sausage string" of potential hypothesis is generated as the output of this process. This output is structured such that the words which are competing at a particular time in the lattice are clustered together to produce "confusion sets". Within each of these sets, the posteriors of each unique word are normalised such that they sum to one over all competing words. Given this representation, a CN hypothesis may be generated by selecting the word with the highest posterior score from each confusion set within the sausage string.

This CN hypothesis is considered to be an *alternative hypothesis*, in the sense that it is the output of an alternative system or process to that of the underlying recogniser for which recognition lattices are available for feature extraction. If the confidence estimation task is that of assigning new confidence scores to words in this hypothesis, the CN posteriors are readily available and may be used as predictor features. These CN posteriors are therefore added to the set of predictor features extracted from the recognition lattices for the CN hypothesis, which are obtained using the hypothesis injection technique proposed in this work (see Section 6.1.2). These posteriors embody significantly different information from that of the LAPR posteriors extracted from the lattice. This is primarily a result of the CN clustering and alignment technique, along with the heuristics associated with this process. These factors have the result that the CN posteriors are effectively computed over different sets of competing hypotheses from those considered when computing the LAPR posterior score.

7.1.4 *Levenshtein Alignment Feature*

A feature of particular relevance when estimating confidence scores for alternative hypotheses is whether or not the lattice 1-best hypothesis matches this alternative hypothesis. If these different hypotheses are in agreement for a given word, the additional evidence in support of this word should indicate that it is more likely to indeed be correct. A Levenshtein alignment is therefore carried out on the "reference" alternative utterance hypothesis $\mathbf{A}$ and the lattice 1-best hypothesis $\mathbf{H}$. The word-level Levenshtein alignment (LA) feature for a particular word i in the reference hypothesis A_i and word A_i in the 1-best hypothesis is then obtained by assigning the discrete value 1 when the hypotheses are aligned, and 0 otherwise, using the Kronecker delta function as follows:

$$\mathrm{LA}(A_i, W_i) = \delta(A_i, W_i). \tag{7.15}$$

7.2 Experimental Setup : ASR Systems and Data

In confidence estimation, the task is that of generating estimates of confidence for the output of a specific underlying recogniser. The details of this recogniser therefore have a significant impact on the confidence estimation process. In this work, ASR systems which comprise the 2010 Cambridge Arabic STT (speech-to-text) system (Diehl *et al.* 2011; Tomalin *et al.* 2010) were used as candidate systems for which the confidence estimation task could be investigated. The aim being that of improving the accuracy of confidence scores for the output of these systems.

The Arabic STT (speech-to-text) system was developed as part of the DARPA GALE program. The front-end processing applied to the audio in this system results in a 39-dimensional set of PLP features (Hermansky 1990; Liu *et al.* 2003) per frame. This feature vector is obtained by extracting 13 PLP cepstra (0^{th} order (energy) to 12^{th} order coefficients) from the audio, and additionally computing the first, second and third order derivatives (or deltas) for each. Thereafter, an HLDA projection (Kumar 1997; Liu *et al.* 2003) is subsequently applied to these features to reduce the dimensionality of the vector from 52 down to 39. The acoustic training data used in developing this system consists of 1538 hours of audio, which comprises data from the broadcast news (BN) and broadcast conversation (BC) domains. A total of 1.2G words was used as language model training data. The vocabulary used is extracted from this data, and consists of the 350k most common words. Both phonetic and graphemic systems were

developed as part of the overall Cambridge Arabic STT system, in which system combination is used to combine multiple different recognisers (or "branches" in the decoding setup). In this work, a particular graphemic recogniser from this set was used. A total of 36 graphemes are used by this system. The decoding structure of the recogniser consists of three consecutive decoding passes, referred to as P1, P2 and P3. The P1 phase is a fast-decode based on gender-independent (GI) acoustic models. The acoustic model applied in this phase is a graphemic word-based model, consisting of 9k tied states. This model is also trained discriminatively using MPE (Povey and Woodland 2002), and employs diagonal covariance matrices. In the next decoding phase, P2, speaker-adapted gender-dependent (GD) graphemic models, which are trained using the supervision generated by the P1 phase, are applied. During this phase, lattices are initially generated using a trigram language model. These lattices are subsequently expanded with a 4-gram language model. The larger lattices resulting from this process are rescored with additional language models. Rescoring is carried out using both a class-based language model, and a neural network language model. In the P3 stage, acoustic rescoring is carried out on the lattices output from the P2 stage. Here, a different set of GD graphemic models are used for acoustic rescoring. These models are obtained by carrying out 1-Best constrained maximum likelihood linear regression (CMLLR) (Gales 1998) and lattice-based MLLR (Padmanabhan *et al.* 2000). Confusion network (CN) clustering (Mangu *et al.* 2000) is subsequently performed on the lattices resulting from this process to produce CN hypotheses.

In general, the output lattices from the P2 phase are more dense than those generated in the P3 stage (having been pruned to a lesser extent). These lattices represent a larger hypothesis space, and ultimately tend to yield more reliable estimates for predictor features. This is a result of the fact that less of the useful competing hypothesis information is pruned away in producing these lattices. The P2 lattices are therefore used as the source of lattice-based predictor features in this work.

A number of datasets employed during the development of the aforementioned recognisers were used in this work. Namely, the dev10d, dev10r and dev10c subsets of the 2010 GALE development data, a subset of the 2009 GALE development data (dev09sub), and the non-sequestered portion of the 2009 GALE evaluation data (eval09ns). The dev10d subset consists of difficult (i.e. high error rate) portions of the data, while the dev10r data has a more typical error rate distribution. The dev10c dataset is one in which particular care was taken in producing the reference transcriptions. The recognisers yield

a range of different error rates on these datasets. For the confidence estimation framework proposed in this work, the dev10c, dev10r and dev09sub datasets were combined to form a training dataset. The dev10d and eval09ns datasets were held out and used for evaluation of the confidence estimation systems. The size of these datasets, and the performance of the underlying recogniser on these datasets is summarised in Table 7.1.

Dataset	Hours	Utterances	P2 1-B WER (%)	P3 CN WER (%)
dev10c+dev10r+dev09sub	27.5	9700	23.6/15.2/19.7	-
dev10d	18.5	7609	30.8	27.3
eval09ns	6.5	1554	13.1	11.1

Table 7.1 *Details of the size of the datasets used, as well as the WER performance of the recogniser used in this work on each dataset. Performance is quoted for both the* 1*-best P2 hypotheses, and the CN P3 hypotheses.*

Each of the subsets of the training dataset were scored using the NIST `SCLite` evaluation software. The resulting output was parsed to construct a parallel corpus of the words, their associated predictor features, and the corresponding `Correct` or `Incorrect` labels for each word hypothesis. In evaluation of the CRF-based models, the updated confidence scores are aligned with the relevant words of the ASR hypotheses. This is output in the standard `MLF` format employed by `HTK` (Young *et al.* 2006), which are subsequently scored using the `SCLite` evaluation pipeline. One of the standard evaluation metrics output by this software is the normalised cross entropy (NCE) score.

7.3 Experimental Setup : CRF Parameter Estimation

The parameters of all CRF models developed in this work are estimated using L-BFGS (Liu and Nocedal 1989), which is a gradient-based optimisation technique. Convergence of the model is considered to have occurred when the improvement in the training data log-likelihoods for successive training iterations is of the order of 10^{-8}, at which point training ceases. The maximum number of iterations is however also restricted to 800, a stopping criterion which is rarely reached in the experiments of this work. A spherical Gaussian weight prior (Goodman 2004) (also referred to as a euclidean or L2 regularisation term) was included in the log-likelihood expression for the model, so as to guard against

overfitting. This term is expressed as:

$$l_2 = \frac{\lambda^2}{2\sigma^2}. \tag{7.16}$$

Various settings for σ were experimented with, which resulted in slight differences in the training data log-likelihood at convergence. However, these parameters did not have a significant impact on performance in evaluation for the confidence estimation task. This effect was also observed in Sutton and Mccallum (2006) when varying σ. A σ value which resulted in middling log-likelihood at convergence was therefore selected for use in training all models.

7.4 Experiments

In the sections which follow, various word-level CRF-based confidence estimation systems will be presented and contrasted against one another. The primary dimensions along which experimentation is carried out are the following:

1. The set of input features (predictor features) used by the system to inform the confidence estimation process.

2. The type of feature functions applied to various predictor features.

3. The class of CRF model used. The main distinction being between that of standard linear chain-CRFs and those that incorporate hidden states.

4. The confidence estimation task itself. This may for instance be that of marking up either the 1-best or alternate hypotheses with confidence scores.

The parameters of various models which are specified through the definition of the aforementioned aspects are estimated using the training dataset. These models are subsequently evaluated on the two held-out datasets (dev10d and eval09ns1), using a variety of metrics. For some the confidence estimation systems which yield important contrasts, detection error trade-off (DET) curves are used as another means of evaluation. The statistical significance of these systems is also analysed.

One of the metrics used in evaluation is the NCE score. Whilst these scores are useful for evaluating confidence estimation performance, improvements in this metric are difficult to relate to the impact

this has in practice. An additional error-based evaluation is therefore carried out for the systems which are shown to yield notable improvements in performance. This analysis is intended to add weight to the stated improvements in performance achieved by these systems, which are contributions of this work.

The error-based analysis is motivated and described from the perspective of a typical setting for the application of confidence scores. Here, the task is that of deciding whether to accept a given word hypothesis as being correct or not. This decision can be made based on whether the confidence score is above or below a particular threshold. Above this threshold level, all hypotheses are accepted as having truly occurred, with those below the threshold being deemed to be incorrect. This decision process results in two types of errors, *false alarms* and *misses*. False alarms occur when a hypothesis accepted as being correct, when in fact this hypothesis is incorrect. Conversely, misses occur when a hypothesis is considered to be incorrect, when it is in fact correct. In analysing system performance through the use of detection error trade-off (DET) curves, this threshold is varied over a wide range of thresholds. However, it is also useful to have a single number as a representation of this error, which can be easily compared across systems. This implies that a single threshold value must therefore be chosen. Given that the confidence scores are probabilities defined on the interval $(0, 1.0)$, one sensible threshold to use is 0.5. Given the aforementioned decision criterion based on this threshold, the number of resulting errors (misses and false alarms) may be determined. An error rate is then computed by normalising this count by the total number of word hypotheses.

In Section 6.5.2, a framework for evaluating the statistical significance of NCE-based results for confidence estimation systems was presented. In order to support key findings in terms of NCE improvements reported in this word, this technique is applied to evaluate the output of selected systems to verify that the differences in performance observed are indeed statistically significant. The criterion applied for the significance test in these experiments is such that the 99.9% confidence level is desired in order to accept systems as being different. This corresponds to a threshold on the p-value of 0.001 (i.e. $\alpha = 0.001$), below which the difference between the results of the systems tested are considered statistically significant.

7.4.1 Baseline Experiments

In Section 6.5.3, a decision tree-based piecewise linear mapping approach was discussed, whereby word posteriors extracted from the lattice (such as the LAPR predictor feature) are re-mapped to produce confidence scores. This approach is employed as the primary baseline for word-level confidence estimation experiments in this work. As a first step in developing an improved confidence estimation system, the performance of the proposed CRF-based confidence estimation models that have similar configurations to that employed by the baseline are evaluated. It should be noted that the resulting CRF models are thereby effectively constrained such that the power of the models to combine multiple predictor features effectively is not exploited.

Given that the single predictor feature used for these evaluations (LAPR) is continuous, it must be represented adequately such that the CRF model can make effective use of this information. There are various techniques that may be applied to represent continuous features, a selection of which are implemented within the CRF framework developed in this work. The efficacy of these various representations, and their impact on performance in reaching (and surpassing) baseline performance are presented in Table 7.3 on the next page. A summary of the feature functions relevant for these experiments is provided in Table 7.2.

Feature Function	Description
M1	First order moment feature
M2	Second order moment feature
QB N	Quantisation with binning, N bins
QBM1 N	Quantisation with first order moment of bins, N bins
Spline NE	Spline approximation, N knot intervals with uniform segmentation
Spline NA	Spline approximation, N knot intervals with uniform occupancy

Table 7.2 *Abbreviations and short description of various feature functions used for representing continuous predictor features.*

Improvements in NCE performance over the naïve baseline (in which the empirical accuracy of the recogniser is taken to be the confidence score for every word), are depicted by values greater than 0, and an ideal system would yield an NCE score of 1. Improvements are therefore signified by increases in the NCE score towards 1. The UMAD metric is a representation of the deviation of a system from the ideal targets for each word (1 or 0), evaluated over utterances. As such, improvements in performance

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	LAPR	0.155	15.60	0.266	7.29
	Baseline	0.325	12.72	0.359	6.91
①	CRF : M1	0.307	11.57	0.325	6.95
②	CRF : M1 + M2	0.325	11.43	0.346	6.74
③	CRF : QB 8	0.309	11.54	0.330	6.87
④	CRF : QBM1 8	0.327	11.35	0.346	6.74
⑤	CRF : Spline(8E)	0.342	11.22	0.359	6.65
⑥	CRF : Spline(8E)+M1+M2	0.342	11.22	0.359	6.65

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
①	Baseline	0.0085	0.0001	60.61

Table 7.3 *Results of experiments using various feature functions to represent the continuous lattice posterior LAPR, contrasted against baseline performance using this single feature. The UMAD(%) and NCE scores on dev10d and eval09ns are shown. Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNCE = average difference between NCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

are signified by decreases in the UMAD score, with an optimal classifier having a UMAD performance value of 0.

As can be seen from the results in Table 7.3, using the LAPR posterior feature without any mapping (i.e. not using a CRF or decision tree) yields poor performance. It is clear that this feature in its raw form is however still indicative of the likelihood of the word being correct, as the NCE scores are greater than 0. Moving to the simplest CRF-based model (system 1) using only the first order moment feature functions improves word-level NCE performance over the unmapped scores, but does not compete with the baseline system using a piecewise linear representation of the feature distribution. This is not surprising, as only modelling the mean is naturally a less expressive representation than an approximation of the distribution. However, it is interesting to note that the sequential nature of the model has a positive impact when considering sequences of words, as the UMAD score is improved by more than 1% absolute on dev10d. Including the second-order statistics in this model (system 2) results in

NCE performance equivalent to the baseline on dev10d, but not on eval09, with gains on both datasets for the UMAD metric. In moving to a quantised representation in system 3 (using the same number of quantisation intervals as the baseline), the simple binning approach which does not learn the mean values of each bin performs slightly better than the model with first order moments. This approach is effectively a piecewise constant mapping, where the constant on every interval is equal to the value at the mid-point of that interval. Including the moments for each bin, which is a mean approximation to a piecewise linear representation (system 4), results in comparable performance to the system using first and second order statistics. Making use of a spline approximation with the same number of intervals as the baseline (system 5) yields NCE performance on dev10d that surpasses the baseline, and is equivalent on eval09, with large improvements in the UMAD metric. This system is compared with the baseline in order to determine whether this result is statistically significant, with the results of this test shown at the bottom of Table 7.3 on the preceding page. The p-value achieved in this test is below the threshold value of 0.001, proving statistical significance of the result. In addition, the snippet improvement ratio (SIP) shows that in achieving the average improvement in NCE of 0.0085, the CRF system outperforms the baseline in 60.6% of the snippets in the evaluation. Finally, the combination of the spline feature functions with the moment feature functions (system 6) yields no performance improvements. This is a result of the fact that the splines are already a more expressive representation than the moment feature functions.

The error rate (in terms of misses and false alarms) was evaluated at a threshold value of 0.5 on the combined dev10d and eval09ns output for the baseline system and the CRF-based mapping with spline feature functions (system 5). The relative reduction in error rate achieved by the CRF-based approach is 1% (from an error rate of 18.8%), which is attributed primarily to the improvement in NCE of 0.17 on dev10d.

The analysis of these results shows that even when the CRF-based approach is limited to using a single feature in isolation, this approach is capable of yielding confidence estimates that are more accurate than those of the baseline system. The sequential nature of the model also contributes significantly to the utterance-level accuracy of the scores, as was made evident through the improvements in UMAD. Finally, it is clearly important to use representations of continuous features within the CRF models which capture more information relating to the distribution of continuous features than stand-

ard first and second order statistics.

7.4.1.1 Modelling Consecutive Errors

In order to investigate the nature of the improvements achieved using the CRF-based approach, a sample transcribed Arabic utterance from the dev10d data is considered. Word-level confidence scores are associated with each word in the utterance, and shown in Figure 7.1. These confidence scores are obtained from systems for which results are provided in Table 7.3 on page 103.

Ref:	ان	ان	ونكي	الءنوان	له	ثالث	مرجءيات	ذيفنت	اقفاق
Hyp:	ان	ان	ونكي	–	الءنوانه	تلقن	ةيءجرم	ذيفنت	اقفاق
Aligned:	C	C	C	D	I	I	I	C	C
LAPR:	0.69	0.97	1.0	–	0.19	0.17	0.79	1.0	1.0
C/I @ 0.5:	C	C	C	–	I	I	C	C	C
Baseline:	0.57	0.91	0.97	–	0.21	0.20	0.65	0.98	0.98
C/I @ 0.5:	C	C	C	–	I	I	C	C	C
CRF:	0.72	0.95	0.94	–	0.11	0.07	0.42	0.96	0.98
C/I @ 0.5	C	C	C	–	I	I	I	C	C

Figure 7.1 *Example of confidence scores assigned to part of an Arabic utterance from the dev10d data. The scores shown are the unmapped LAPR posterior, the baseline system scores, and scores generated by the CRF model applied to this single feature. The reference and hypothesised word sequences are shown, along with the alignment information for these sequences. The assignment of each word as being correct ("C") or incorrect ("I") based on a score threshold of* 0.5 *is also shown. The label "D" indicates a deletion error.*

The reference and hypothesised sequence of words for the utterance in the audio are shown, and the result of performing an alignment between these sequences is also provided. This alignment indicates whether the hypothesised word is truly correct (depicted as "C"), incorrect (depicted as "I"), or whether there is a missing word in the hypothesis (i.e. a deletion, depicted as "D"). The resulting "C" and "I" labels can be considered as the ideal labels that should be assigned to the words hypothesised by the ASR system. The first set of confidence scores shown for each word in Figure 7.1 are the unmapped LAPR predictor features. These predictor features are also used by the decision tree baseline and a CRF model with spline feature functions (i.e. system 5 in Table 7.3 on page 103) to produce competing confidence scores. For the purposes of illustration in this example, a threshold of 0.5 is applied to these confidence scores as a means of obtaining "C" and "I" labels for each word.

Considering that the confidence scores should be closer to 0 for incorrect words, and closer to 1 for correct words, it is clear to see that the unmapped word posteriors (LAPR) show some correlation with these ideal values. However, they are not entirely accurate, and in the case of the third incorrect word for instance, this posterior score is above the threshold value of 0.5. Applying the thresholding decision to this score will therefore result in an error, as the word will be assumed to be correct. This fact also holds for the baseline score mapping. The CRF model uses the exact same input, but makes use of transition feature functions (of the form defined in Section 4.2.3) to consider an entire sequence of labels. This transition structure allows the model to capture the fact that errors tend to occur in consecutive words. As there is a run of two consecutive incorrect words before the third incorrect word, the CRF model is able to account for this fact and accordingly reduce its confidence in this word being correct. This occurs despite the relatively high word posterior score provided as input to the model. The resulting confidence score is below the threshold in this case, resulting in the correct label (i.e. "I") being assigned to this word. Furthermore, it is seen that for each word in the sequence of consecutive incorrect words, the CRF-based mapping yields confidence scores which are consistently lower than the other systems, and closer to the ideal value of 0. This example serves to support the results in this section which have shown that the sequential nature of the CRF model can be exploited to improve CE performance.

7.4.2 Spline Feature Functions for Continuous Predictor Features

As was shown in the results of Section 7.4.1, the use of spline feature functions enables the CRF model to improve over baseline performance using a single predictor feature. The number and placement of spline knots within the approximation are however important meta-parameters of these feature functions, and therefore warrant further investigation. The results of experiments using various configurations for the spline parameters are shown in Table 7.4 on the next page.

The "MaxEnt" systems (1 and 2) presented in Table 7.4 on the facing page correspond to a configuration in which the transitions between output state labels are disabled in the CRF, effectively reducing it to a maximum entropy model. Such a model is directly comparable with the non-sequential decision-tree based approach, and therefore provides an interesting point of comparison for the spline approximation technique. An approach proposed in this work is that of placing knot points automatically so

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	Baseline	0.325	12.72	0.359	6.91
①	MaxEnt : Spline 8E	0.321	12.63	0.354	6.96
②	MaxEnt : Spline 8A	0.327	12.58	0.361	6.90
③	CRF : Spline 4E	0.334	11.29	0.353	6.67
④	CRF : Spline 8E	0.342	11.22	0.359	6.65
⑤	CRF : Spline 4A	0.346	11.23	0.365	6.61
⑥	CRF : Spline 8A	0.346	11.19	0.365	6.60

Table 7.4 *Results of experiments in which the number and placement of spline knots is varied. Experiments are carried out with non-sequential maximum entropy models, as well as with CRF models. The single continuous predictor feature used is the LAPR score.*

as to enforce a uniform occupancy of datapoints over intervals (denoted with an "A" in the table). This is in contrast to the technique suggested in (Yu *et al.* 2009), in which these knot points are placed evenly so as to yield a uniform segmentation over the range of the continuous predictor feature (denoted with an "E" in the table). The results show that the maximum entropy model with automatically-derived knot placement (system 2), is capable of outperforming the baseline decision tree-based system. It is not surprising that the uniform segmentation approach (system 1) yields slightly inferior performance, as the both the decision tree baseline and automatic spline knot placement approaches make use of boundaries that are effectively optimal in some sense.

When considering the experiments in Table 7.4 where the uniform segmentation approach for splines ("E") is employed by the CRF models, it is clear to see that increasing the number of knots (and consequently the effective number of quantisation intervals) from 4 to 8 yields substantial improvements in both the NCE and UMAD metrics. However, when the automatic approach for knot placement ("A") is applied, the model using splines with 4 intervals (system 5) is able to outperform the model in which the standard uniform segmentation is used over 8 intervals (system 4). It is interesting to note that there is an improvement in the UMAD metric but not in the NCE score in going from 4 to 8 knot points with this regime in systems 5 and 8. The UMAD performance for the CRF systems is significantly improved (lower deviation) over that in the MaxEnt systems, which re-iterates the fact that the sequential approach not only improves word-level NCE performance, but also utterance-level

performance.

The results show that by making use of a data-driven technique proposed in this work, which selects knot points automatically based on uniform occupancy, the number of parameters required by the CRF model to achieve reasonable performance using a single predictor feature may be decreased.

7.4.3 *Predictor Features*

A number of predictor features were investigated in this work. These predictor features are sources of information which serve to inform the confidence estimation process. Whilst one of the main strengths of the CRF-based framework used is that multiple features can be combined effectively, it is interesting to study the suitability of various predictor features for the confidence estimation task in isolation. This is also an integral part of the development of such a confidence estimation system, as predictor features which are not useful may effectively be disregarded or recomputed under different conditions. The results shown in Table 7.6 on the facing page are those obtained using CRF models which have been trained using various predictor features in isolation to estimate confidence scores for 1-best ASR hypotheses. A summary description of the predictor features used is provided in Table 7.5.

Feature Function	Description
SP(e)	Similarity posterior with a maximum edit distance of e
WS(d)	Word stability for d^{th} competing word
ACRATIO	Acoustic model score ratio
LPS(d)	Lattice Posterior Stability for d^{th} competing word
WCR	Word Confusion Ratio
LAS	Lattice Acoustic Stability
AASWP	Average Alternative Sub-Word Posterior
ABASWP	Average Best Alternative Sub-Word Posterior
LALMP	Lattice Arc Language Model Posterior
LAPR($\beta = x$)	Lattice Arc Posterior Ratio, language model scores scaled by x
LAPR($\gamma = x$)	Lattice Arc Posterior Ratio, acoustic model scores scaled by $\frac{1}{x}$

Table 7.5 *Short description of various predictor features utilised in word-level CRF experiments.*

It is clear to see from the results in Table 7.6 on the facing page that various forms of the lattice posterior probability (LAPR) yield the best performance, when considered in isolation. This is not surprising however, as these posterior scores are based on the probabilities calculated during a

	dev10d		eval09	
System	NCE	UMAD	NCE	UMAD
Baseline	0.325	12.72	0.359	6.91
ABASWP	0.010	20.23	0.038	10.31
AASWP	0.057	18.76	0.060	9.95
ACRATIO	0.075	18.73	0.103	9.82
SP(2)	0.117	16.90	0.176	8.62
LPS(1)	0.146	17.15	0.195	8.98
SP(3)	0.156	15.15	0.153	8.55
WS(2)	0.164	16.74	0.194	8.87
WS(3)	0.164	16.29	0.143	9.76
WS(1)	0.169	16.63	0.205	8.83
LALMP	0.197	15.30	0.187	8.67
LAPR($\gamma = 40$)	0.226	13.99	0.254	7.92
LAS	0.235	14.86	0.267	7.88
LPS(2)	0.269	13.35	0.288	7.61
SP(1)	0.283	11.98	0.262	7.33
LPS(3)	0.287	12.55	0.315	7.18
LAPR($\gamma = 10$)	0.304	12.17	0.319	6.95
LAPR($\beta = 1.2$)	0.309	11.93	0.336	6.81
LAPR($\beta = 0.8$)	0.334	11.53	0.350	6.69
LAPR($\gamma = 13$)	0.340	11.26	0.355	6.67
LAPR($\gamma = 14$)	0.342	11.22	0.359	6.65

Table 7.6 *Results of single predictor feature experiments. NCE and UMAD scores are shown for CRFs in which a spline feature function with 8 evenly spaced intervals are applied to a single predictor feature. A number of LAPR predictor features were experimented with using different scaling factors, the value of which is indicated in parentheses where appropriate.*

forward-backward pass over the lattice, and therefore inherently encode some contextual information. Furthermore, the acoustic and language model scores used in the calculation of these probabilities are generated by sophisticated models in a state-of-the-art ASR system. These models are themselves already effective estimators of whether or not a word is likely to be correct.

The remaining predictor features shown in the table do still yield NCE scores greater than 0, which indicates that they are also useful predictors of word correctness. However, as a sole basis for estimating confidence scores, they are effective to a lesser extent than the LAPR predictor feature. The predictor features derived from the sub-word-level alternative recogniser (AASWP) and (ABASWP) have relatively low performance scores when evaluated on the word level. These scores are however above 0

suggesting they do represent some degree of useful information. The better of these sub-word-level predictor features is AASWP. This is the alternative system's score for the sub-words comprising the word hypothesised by the underlying system, and is therefore directly related to the confidence of this word hypothesis. The ABASWP predictor feature on the other hand uses the timing of the sub-words comprising the hypothesised word, but is based on the highest-scoring sub-word from the alternative recogniser on each such interval (effectively an alternative hypothesis). It is therefore to be expected that ABASWP would have a lower degree of correlation with the likelihood of the hypothesised word being correct, as is evident from the results.

Although many predictor features are generally not able to compete with the LAPR word posterior in a single-feature system, it is nevertheless expected that these features should by their nature provide information which is different from the LAPR predictor feature. If such predictor features do indeed encode different aspects of the problem, they should contribute positively in a multi-feature confidence estimation system, resulting in improved confidence score estimates. In order to investigate this effect, each of the predictor features is combined with the best single predictor feature (LAPR). This results in CRF-based systems in which two predictor features are combined to estimate confidence scores, the results for such systems are shown in Table 7.7 on the next page.

The first interesting result in Table 7.7 on the facing page is that combining the two features which performed the best in isolation (LAPR+LAPR($\gamma = 13$)), yields no improvement in performance. This is however not surprising, as these predictor features are highly correlated. These posterior scores have almost identical definitions, with the only discrepancy being a relatively small change in the grammar scaling factor used in their calculation. The nature of the model is such that combining such highly correlated information does however not degrade performance below that of one of the correlated features. This fact is one of the motivating factors for the use of CRF models for the confidence estimation task.

The two-way feature combination which yields the best performance is the combination of the LAPR and LALMP predictor features, which results in a 3.8% relative improvement in NCE on dev10d. This suggests that there must be some instances where the language model scores (and the related posterior feature LALMP), represent significantly different information from the overall posterior score, which includes the acoustic information. Using the acoustic stability metrics (LAS and DEV), also res-

	dev10d		eval09	
System	NCE	UMAD	NCE	UMAD
Baseline	0.325	12.72	0.359	6.91
LAPR+SP(1)	0.343	11.19	0.360	6.64
LAPR+LPS(1)	0.345	11.23	0.363	6.69
LAPR+LAPR($\gamma = 13$)	0.342	11.22	0.360	6.64
LAPR+LAPR($\gamma = 15$)	0.342	11.22	0.361	6.63
LAPR+WS(1)	0.345	11.23	0.363	6.69
LAPR+LPS(2)	0.345	11.20	0.362	6.64
LAPR+ACRATIO	0.347	11.15	0.363	6.62
LAPR+LPS(3)	0.347	11.16	0.365	6.61
LAPR +WCR	0.349	11.21	0.368	6.55
LAPR+LAS	0.349	11.25	0.369	6.56
LAPR+ABASWP	0.349	10.97	0.361	6.58
LAPR+AASWP	0.354	10.80	0.364	6.56
LAPR+LALMP	0.353	11.04	0.366	6.56

Table 7.7 *Results of two-way predictor feature combination experiments. NCE and UMAD scores are shown for CRF models in which spline feature functions with 8 evenly spaced intervals are applied to each predictor feature.*

ulted in improvements over the single feature configuration. The same trend is observed when using the "confusability" ratio predictor feature (WCR). This suggests that the number of competing paths kept active during recognition is useful alternate information to the posterior scores. The predictor features based on an alternative sub-word-level recogniser also yielded decent improvements, with the AASWP predictor proving to be the better of the two. In fact, the best UMAD results on dev10d were obtained in the systems using the sub-word-level features, with a relative improvement of up to 2.1% over the next best two-way combination of features. A recurring theme in contrasting the results in Table 7.7 with those in Table 7.6 on page 109, is that predictor features which do not perform especially well in isolation, often seem to contribute more in combination with other predictor features.

7.4.4 Combination of Multiple Predictor Features

As was made evident through analysis of the results presented in Table 7.7, some predictor features contribute more to the overall confidence estimation process than others. Perhaps more importantly, those that don't necessarily represent sufficiently new information when combined with a similar fea-

ture, do not degrade the performance of the overall system. This is a favourable characteristic of the CRF-based framework, which makes it possible for one of the primary advantages of this approach to be exploited. This advantage is that of the ability to effectively combine multiple (more than two) predictor features, without needing to be concerned that doing so might directly result in a degradation in performance. Feature combination results in more sophisticated, complete models for confidence estimation, which are capable of achieving higher levels of performance. It should however be noted that including additional predictor features in the model does result in an increased computational cost, as additional parameters must naturally be estimated for these features. As such, despite the fact that the model is insensitive to negative effects of multiple correlated features, blindly including a very large number of predictor features in the model is certainly by no means an optimal approach. Experiments were carried out in which different sets of predictor features were combined to produce larger systems, the results for a selection of which are presented in Table 7.8 on the facing page.

In the results shown in Table 7.8 on the next page, the "BEST10FEAT" configuration consists of the 10 predictor features (including LAPR),which yielded the best performance in two-way combinations with LAPR. This set of predictor features therefore includes LAPR, WCR, LAS, ALMP, LPS(3), ACPOST, LPS(2), WS(1), LPS(1) and SP(1). Considering the results for the "8E" systems, including 8 more features in the "BEST10FEAT" set (system 5) over the best two-way combination of system 2 (LAPR+LALMP) yields improvements in both NCE and UMAD, with the margin being slightly higher on eval09ns than on dev10d. This incremental improvement seems to indicate that the amount of new information in the additional predictor features is somewhat limited. It is also evident that the trend observed on a single feature in using a uniform occupancy-based spline placement technique holds in systems within which multiple predictor features are used. It should be noted that in the case of multiple predictor features, the locations of the knot points are calculated and stored separately for each resulting dimension of the input feature vector, such that a global knot placement is not applied. The best system shown at the bottom of Table 7.8 on the facing page (system 9) is one in which the complete set of predictor features available on the word level is utilised. This system yields a sizeable relative improvement of 11.4% and 5.3% in NCE over the decision tree baseline using the LAPR predictor feature. In comparison to a CRF system using the LAPR feature in isolation (system 1), this full-featured system still shows large gains on dev10d of 8.5% with the performance improvement on

System		dev10d		eval09ns		
Num	Description	NCE	UMAD	NCE	UMAD	#Parms
	Baseline	0.325	12.72	0.359	6.91	–
①	LAPR 8E	0.342	11.22	0.359	6.65	34
②	LAPR+LALMP 8E	0.353	11.04	0.366	6.56	51
③	BEST10FEAT 4E	0.356	11.08	0.372	6.53	115
④	BEST10FEAT 4A	0.361	11.56	0.376	6.55	115
⑤	BEST10FEAT 8E	0.359	11.03	0.374	6.51	195
⑥	BEST10FEAT 8A	0.362	10.95	0.376	6.47	195
⑦	⑥ +AASWP+ABASWP 8E	0.367	10.75	0.376	6.46	232
⑧	ALLFEAT 4E	0.367	10.71	0.378	6.40	516
⑨	ALLFEAT 8E	0.371	10.59	0.378	6.37	807

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
①	Baseline	0.0085	0.0001	60.6
⑨	Baseline	0.0330	0.0001	82.1
①	⑨	0.0240	0.0001	82.1

Table 7.8 *Results for experiments in which multiple predictor features are combined. A set which comprises the* 10 *best feature functions is denoted by "BEST10FEAT", and the complete set of all continuous predictor features is denoted by "ALLFEAT". NCE and UMAD scores are shown for CRFs in which spline feature functions are applied to each of the predictor features. The number of parameters comprising each model is also shown. Results of significance tests on selected systems are also shown in the bottom table, which are evaluated on the combined dev10d and eval09ns datasets. ΔNCE = average difference between NCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

eval09ns remaining unchanged at 5.3%. This system (9) is compared with the baseline in order to assess the statistical significance of these results. The results of this significance test evaluation for the full-featured system and the single-feature LAPR system (1) are shown at the bottom of Table 7.8. The p-value achieved by the full-featured system is below the threshold value of 0.001, proving statistical significance of the result. It is also observed that this system outperforms the baseline on 82.2% of the snippets in the audio, which is 21.6% absolute more than the single-feature system. This analysis therefore serves to prove that these systems developed in this work do not only show improved per-

formance over a competitive baseline, but also result in systems which are consistently different (and by extension consistently better) than this baseline. These results provide additional evidence in support of including a number of additional predictor features in the model. Although this full-featured CRF model consists of more than 20 times the number of parameters than in the aforementioned single-featured CRF model, training and inference in this model is not prohibitively expensive in terms of time complexity using the CRF framework developed as part of this work.

In order to highlight the improvements in performance achieved by combining multiple predictor features in the CRF-based model further, the error rate-based evaluation previously discussed was carried out. The error rate (at a score threshold of 0.5) for the best combined system is compared with that of the baseline on the combined dev10d and eval09ns output. The relative reduction in error rate achieved by the CRF-based approach is 5.9%, which is almost 5% absolute more than that achieved in the single-featured system.

7.4.4.1 Analysis of Detection Error Trade-off Curves

The analysis of system performance thus far has focussed on single metrics which either represent the degree of information in confidence scores (NCE), or their deviation from the ideal scores (UMAD). The performance of confidence estimation systems may also be analysed with the aid of detection error trade-off (DET) curves, which are described in Section 3.4. These curves illustrate the performance of systems over a range of effective operating points. In order to gain further insight into the performance of the CRF-based confidence estimation models, DET curves are plotted and shown for a selection of systems on the eval09ns and dev10d datasets in Figures 7.2 on the facing page and 7.3 on page 116 respectively.

It is evident from the plot in Figure 7.2 on the facing page (corresponding to the eval09ns dataset), that using a CRF-based mapping of the LAPR predictor feature, performance is improved over the decision tree baseline mapping for the majority of operating points (except those corresponding to low false alarm probabilities). The improvements are particularly evident when considering the lower right part of the plot, which corresponds to lower miss probabilities (i.e. lower decision thresholds). In this region, the scores in the baseline system reach a point where the false alarm probability continues to increase with no reduction in miss probability. The LAPR CRF system yields noticeable improvements

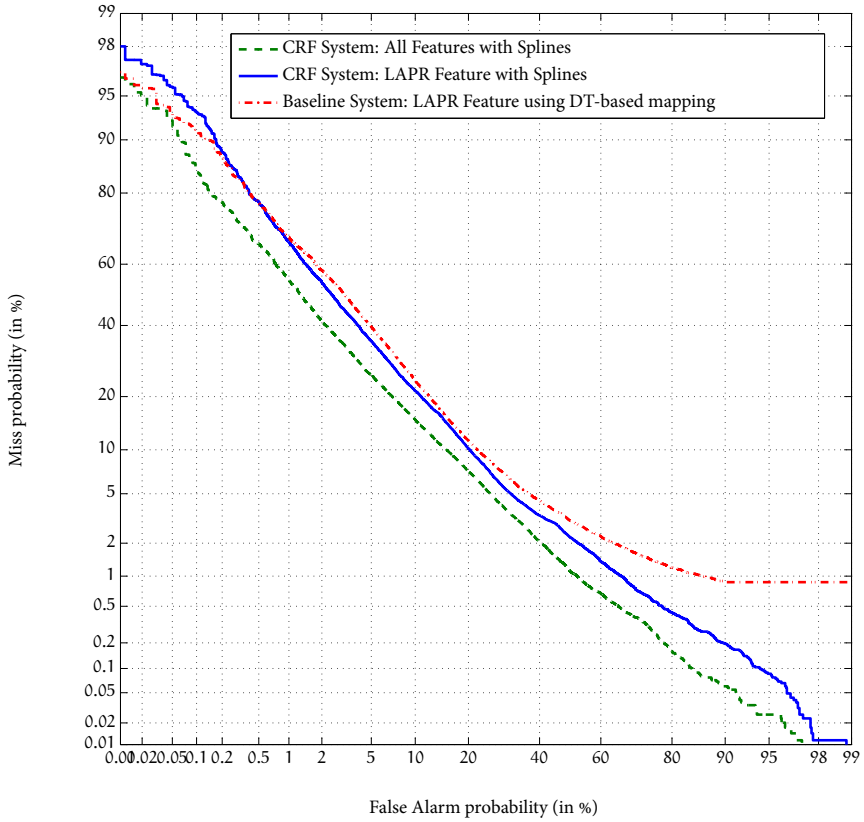


Figure 7.2 *DET curves showing the performance of CRF-based confidence estimation systems in which multiple predictor features are combined, evaluated on the eval09ns dataset. Curves for the baseline system and a single-feature CRF are included in the plot for comparison.*

well before the baseline performance reaches a plateau, with an improvement of 0.8% absolute in miss probability at a false alarm probability of 90%. Similar differences in performance are observed between these two systems when considering the results for the dev10d dataset, shown in Figure 7.3 on the next page. On this dataset the absolute reduction in miss probability at 90% false alarm probability is 0.6%. Performance of the LAPR CRF at low false alarm probabilities is similar to the baseline for this dataset. There is therefore a discrepancy between this result and the poor performance in this region seen on eval09ns. The LAPR CRF model may therefore be overestimating the confidence to some extent for particularly high values of the LAPR predictor feature, which are more common on the lower error-rate eval09ns dataset.

The full-featured CRF system shows improved performance over the single feature (LAPR) CRF,

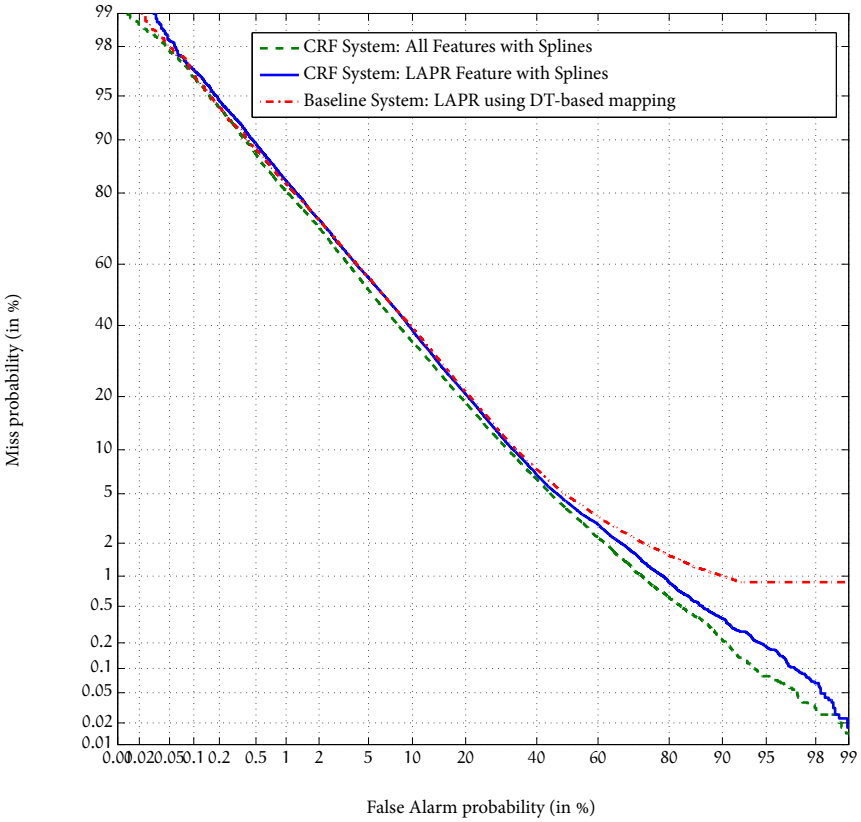


Figure 7.3 *DET curves showing the performance of CRF-based confidence estimation systems in which multiple predictor features are combined, evaluated on the dev10d dataset. Curves for the baseline system and a single-feature CRF are included in the plot for comparison.*

and shows a consistent, clear improvement over the baseline at all operating points. This system does not perform poorly at low false alarm rates, as is the case for the single feature CRF. This implies that this system does not overestimate to the same extent, and yields improved confidence estimates for word hypotheses which are likely to be correct. In comparison with the baseline system, there is a clear separation between the two curves, indicating that performance is indeed improved to a large degree over all operating points. As an indication of this, the absolute improvement in miss probability at false alarm rates of 0.1% and 10% is 10%, and 0.95% at a 90% false alarm rate. At a miss probability of 1% the absolute improvement in false alarm probability is considerably large at 28%, with a miss probability of 10% corresponding to an absolute improvement of 8%. A more modest improvement in false alarm probability of 0.1% absolute is observed at a miss probability of 90%. This is however to be

expected, as at high miss probabilities a large proportion of the word hypotheses are being discarded, and only those that are highly likely are retained. Such hypotheses are likely to get high (accurate) confidence scores from both systems. The large gains observed at higher false alarm probabilities (or lower miss probabilities) are encouraging results, as they show the utility of the approach in improving performance in regions where the recogniser is unsure of its hypotheses.

*7.4.5 A Long-range Predictor Feature : TF*IDF*

Predictor features based on the recognition process at the time a hypothesis is made are good indicators of whether the word is correct, and by extension, good predictor features for use in confidence estimation. However, such features do not inherently incorporate information of a long-range nature. In this work, the TF*IDF scores commonly applied in the information extraction literature are used as a means to investigate the effect of including such long-range information. The results for systems which make use of the TF*IDF scores are presented in Table 7.9.

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	Baseline	0.325	12.72	0.359	6.91
①	LAPR	0.342	11.21	0.359	6.65
②	LAPR+TF	0.343	11.29	0.363	6.64
③	LAPR+IDF	0.345	11.24	0.357	6.89
④	LAPR+TF*IDF	0.339	11.34	0.362	6.65
⑤	LAPR+TF+IDF+TF*IDF	0.352	11.00	0.362	6.81
⑥	BEST10FEAT+AASWP+ABASWP	0.367	10.75	0.376	6.46
⑦	⑥ +TF+IDF+TF*IDF	0.371	10.73	0.378	6.57

Table 7.9 *Results for systems in which the TF*IDF predictor features are used. NCE and UMAD scores are shown for CRFs in which spline feature functions with 8 evenly spaced intervals are applied to each of the predictor features. A set of the* 10 *best lattice-based feature functions is denoted by "BEST10FEAT".*

The results in Table 7.9 show that combining the term frequency (TF) with the LAPR predictor feature (system 2) results in improvements in both the NCE and UMAD performance metrics. Including the inverse document feature (IDF) in system 3 yields slightly larger gains over system 1 on dev10d, in

terms of both the NCE and UMAD metrics. This trend is however reversed on the eval09ns dataset. Combining the TF*IDF predictor feature with the LAPR predictor feature (system 4) yields improvements over the LAPR-only system on eval09ns, but with a degradation on dev10d. However, when these predictor features are combined (system 6), the model is able to exploit this complementary information to outperform the baseline by a decent margin. This is particularly true on dev10d where a relative improvement in NCE from 0.342 to 0.352 of 2.9% is achieved.

The combination of these long-range features with the set of lattice-based features investigated in the multiple feature combination systems detailed in Section 7.4.4 is also investigated. The TF*IDF features are used in addition to the best word-level lattice-based predictor features ("BEST10FEAT"), and the averaged sub-word-level scores from an alternative recogniser (AASWP and ABASWP). The resulting system (7) shows improvements over a system without the TF*IDF features (system 6). In systems such as this with a relatively large number of predictor features, the gains do tend to be of a more incremental nature. However, the improvements show that predictor features of a long-range nature, which are not extracted from the ASR system can contribute to the confidence estimation process, and this is still true in a larger system.

7.4.6 Hidden-state CRF

An observation based on the experimental results in Section 7.4, is that the sequential nature of the CRF models is well suited to the task of confidence estimation, and can be exploited to yield improved confidence scores. This is partly due to the fact that the problem itself is a sequential one, in which scores must be assigned to words within a word sequence, referred to as an utterance. The sequential aspect of the CRF models acting on the word level implies that the "dynamic" characteristics of the data can be captured, particularly that of transitions from one word to another word (which may both have either a `Correct` or `Incorrect` label). However, it is reasonable to assume that there are some other aspects of the problem which correspond to a dynamic transition structure at a different level of granularity from that represented explicitly by the model (i.e. consecutive word transitions). The use of CRF models with hidden states allows for such hidden structure, at a level of granularity which is effectively not observed as part of the word-level state sequence, to be captured implicitly by the model. The motivation for the use of these models, is that the hidden structure could correspond to

a soft segmentation of words into sequences of `Correct` and `Incorrect` words. This is in keeping with the idea of regions or "pools" of confidence existing within recogniser output, which is one of the concepts put forward in this thesis.

As was detailed in Section 6.3, the CRF framework developed in this work (`CRFTK`) was extended such that hidden states could be included in the models trained for confidence estimation. When training and developing these models, there are however some considerations which should be mentioned:

1. Parameter initialisation and constraints. In order for the hidden states to learn something meaningful, symmetry must be broken for the hidden state values. This may be achieved by imposing constraints on the hidden state values to enforce some structure (such as the disjoint constraints proposed in Morency *et al.* (2007)). Alternatively (or additionally), the parameters values can be initialised randomly such that they will tend to learn this structure inherently.

2. Caching statistics. In the hidden-state model, the empirical expectancy must be recalculated during each iteration, given the parameters estimated in the previous iteration. The statistics may therefore not be cached between transitions to speed up the training process, as was possible in the case of standard linear-chain CRFs.

3. Complexity. The inclusion of hidden states implies that many additional parameters must be estimated, with summations over all possible hidden state configurations being required when computing the necessary statistics. This fact, coupled with the inability to perform caching, prompted the implementation and use of a parallel training mode to make training such hidden-state models feasible.

4. Number of hidden state values. The number of values that the hidden state variables may take must be defined. The number of values required should be related in some way to the nature of the assumed hidden state space to be modelled. This meta-parameter has a large impact on the complexity of the model, and should therefore be chosen with care.

Results of experiments for various CRF-based confidence estimation systems which include hidden states are presented in Table 7.10 on page 121. The results show that even when only considering a single predictor feature (LAPR), a hidden-state CRF model with 2 hidden state values (system 2) yields

substantial improvements in both the NCE and UMAD metrics over a system with no hidden states. These hidden states are constrained in such a manner that the value of the hidden state may only stay the same or increase during a sequence of `Correct` or `Incorrect` labels. The improvements obtained therefore suggest that the hidden state structure is effectively modelling the desired characteristics pertaining to regions of confidence which exist at a level above that of the word-level. The reduction in the UMAD metric over CRF models with no hidden states also suggests that the hidden-state structure of these models is beneficial for estimating scores for the words present in the "visible" word-level sequence. A further encouraging result is that the performance gain from including hidden states in the single-feature (LAPR) model is greater than any gain that was obtained when combining this feature with any other predictor feature in the CRF with no hidden states. A statistical significance test was carried out for this system (2) in relation to the baseline, with the results shown in the bottom part of Table 7.10 on the next page. These results are shown to be significant, with a p-value of less then 0.001. In addition, the average improvement in NCE of 0.04 results in the hidden-state system outperforming the baseline on 70.3% of the snippets in the audio.

The results of the experiment in which the number of hidden state values was increased to 3 showed modest performance improvements, with a significant increase in model complexity. The fact that the gains are minimal when incorporating additional state values does however suggest that, given the left-to-right increasing constraints that are applied, two states are sufficient to capture the desired additional structure.

Making use of the complete set of lattice-based continuous features as input to the hidden state model, with spline feature functions using 4 knot intervals applied to each predictor feature, yields a complex model consisting of 945 parameters. Fewer spline knots were used in this system to reduce training times. This full-featured hidden-state CRF outperforms the baseline system in terms of NCE by 17.2% and 6.7% on dev10d and eval09ns respectively. This result is shown to be significant (with a p-value of less than 0.001). In addition, this system outperforms the baseline on 81.2% of the snippets. The hidden-state system is also able to outperform the equivalent model with no hidden states. The results show that this system achieves a substantial improvement in NCE on dev10d of 2.4% relative over the linear-chain model using the same predictor features. This result proves to be significant, with the hidden-state system outperforming the comparable linear-chain system on 65.5% of the snippets.

Hidden-state CRF models using many predictor features with hidden states yield the highest word-level performance scores reported in this work.

The improvements in performance observed for hidden-state models is supported by the results of an error rate-based evaluation. The error rate in terms of misses and false alarms for the full-featured hidden-state CRF system is compared with that of the baseline on the combined dev10d and eval09ns output. The relative reduction in error rate achieved by the CRF-based approach is 6.9%, which is almost 6% more than that achieved in the single-featured system, and 1% more than that obtained for the CRF model with no hidden states. The absolute reduction in error rate is 1.3%, and is the result of 2125 fewer errors being produced at the representative operating point chosen for the evaluation (a score threshold of 0.5).

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	Baseline	0.325	12.72	0.359	6.91
①	CRF LAPR 8E	0.342	11.22	0.359	6.65
②	HCRF LAPR 8E H=2	0.357	10.99	0.367	6.62
③	HCRF LAPR 8E H=3	0.357	11.09	0.371	6.60
④	ALLFEATS 4E	0.372	10.62	0.378	6.41
⑤	ALLFEATS 4E H2	0.381	10.46	0.383	6.38

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
②	Baseline	0.020	0.0001	70.3
⑤	Baseline	0.040	0.0001	81.2
⑤	④	0.007	0.0001	65.5

Table 7.10 *Results contrasting the performance of linear-chain and hidden-state CRF models, when using a single feature and the complete set of features. Fewer evenly-spaced spline intervals (4) were used in the full-featured models ("ALLFEAT") than in the single feature model (8). Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNCE = average difference between NCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

7.4.6.1 *Analysis of Detection Error Trade-off Curves*

As in the case of linear-chain CRFs, the performance of hidden-state CRFs is analysed with the use of DET curves to investigate the performance of these systems over a range of operating points. The DET curves for these systems on the eval09ns and dev10d datasets are provided in Figures 7.5 on the facing page and 7.4 respectively.

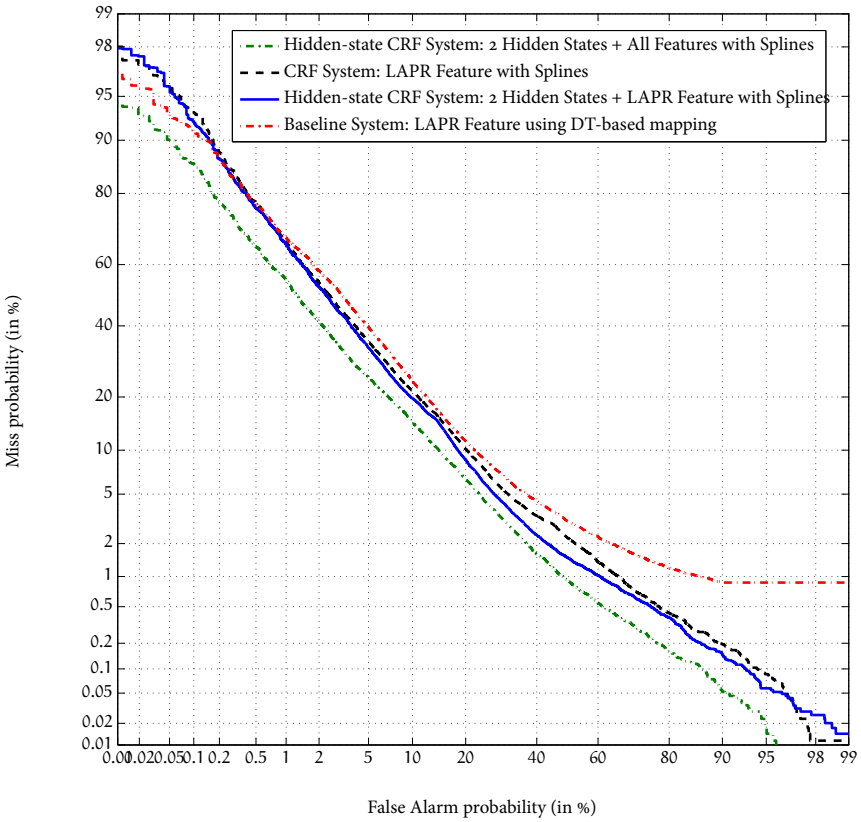


Figure 7.4 *DET curves showing the performance of hidden-state CRF systems over a baseline and linear-chain CRF systems on the eval09ns dataset.*

In Figures 7.4 and 7.5 on the next page, DET curves are shown for single and full-featured hidden-state CRF systems, a linear-chain CRF and that of the decision tree baseline. It is clear that all variants of the CRF-based approach are capable of outperforming the baseline system over a large range of operating points, particularly those at false alarm probabilities greater than roughly 1%. Contrasting the hidden-state LAPR system with the standard LAPR CRF, it is observed that the hidden-state configuration yields larger margins of improvement between false alarm rates of 15% and 80%. Including

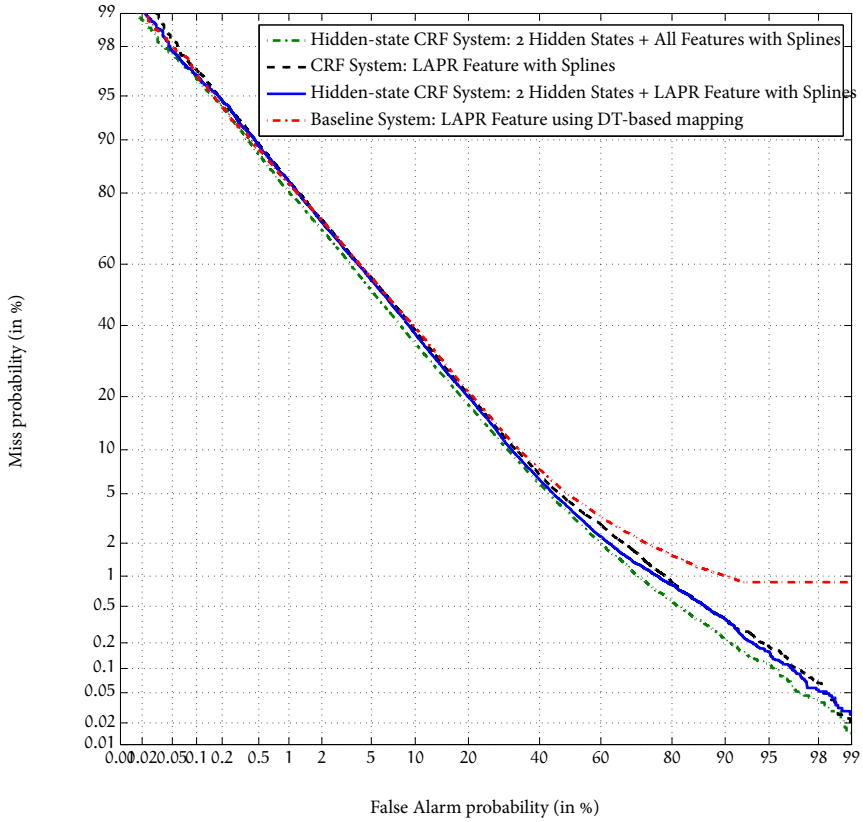


Figure 7.5 *DET curves showing performance of hidden-state CRF systems over a baseline and linear-chain CRF systems on the dev10d dataset.*

hidden states in the single-feature model does not improve the performance for low false alarm rates, where both CRF-based approaches tend to overestimate the confidence.

The full-featured system DET curve yields large improvements in performance over other CRF-based approaches and the baseline. This system is also capable of achieving good performance at low false alarm rates. A similar improvement was obtained in going from a single-feature system to a full featured one without hidden states (see Section 7.4.4.1). This is attributed to the fact that the underlying LAPR predictor feature overestimates the confidence when the words are highly likely. Including additional features in the CRF model has the effect of smoothing the scores, such that the final confidence scores do not overestimate confidence levels for highly likely words.

7.4.7 *Confidence Estimation for Alternative Hypotheses*

Thus far, confidence estimation has been investigated within the context of using features extracted from the underlying ASR system lattices, from which a particular hypothesis (such as the 1-Best), has been selected to be annotated with confidence scores. However, there may be situations where the hypothesis for which confidence scores are desired is not extracted directly from the same lattices. One situation where this is true is in the use of confusion network (CN) hypotheses. After carrying out confusion network clustering on given lattices to produce CN hypotheses, these typically yield improved WER performance over the 1-best (MAP) hypotheses of the lattices. These hypotheses are therefore typically output by the ASR system, and should therefore be marked up with confidence scores.

Techniques for annotating alternative hypotheses (such as the CN hypotheses) with confidence scores are therefore developed in this work. One advantage of these approaches is the convenience of obtaining confidence scores for a hypothesis which may be produced in a manner which does not inherently generate confidence scores, or otherwise produces scores which are inaccurate. Another related benefit, is that this makes it possible for techniques to be applied through which the accuracy of confidence scores for such hypotheses may be improved. This is achieved in this work by exploiting the power of the CRF-based approach to capture sequential aspects of the problem and exploit information from multiple sources. For instance, information from the system which produces a hypothesis to be annotated can be fused with information from the recognition lattices produced by a different system. This "different" system may correspond to a completely different recogniser applied to the same audio, or as is investigated in this work, the output of a different stage of a sophisticated recognition pipeline.

In the task of estimating confidence scores for CN hypotheses, direct comparison can be drawn between the CRF-based approach, and one which uses the confusion network posteriors. These posteriors are a by-product of the clustering process, and are related to word confidence. As such, they are often used as confidence scores and are therefore a good point for comparison. This configuration for confidence estimation shall be referred to as P2-cn.

In order to further investigate the effect of incorporating information from alternate sources, experiments were carried out in which the alternative CN hypotheses to be annotated are from the output of a phase further down the pipeline than that from which the recognition lattices are taken to extract

predictor features (the P2 phase). The hypotheses to annotate are from the word-based P3 system, which is still related to the P2 system, as it is the result of rescoring the P2 lattices. This configuration shall be referred to as P3-cn. The results of these experiments are summarised in Tables 7.11 and 7.12, where the complete feature set considered consists of the LAPR, LAS, LA and CNP predictor features and is denoted as "ALL".[3]

		dev10d		eval09ns	
Sys	Feat	NCE	UMAD	NCE	UMAD
—	CNP	0.134	14.97	0.240	6.89
DT	CNP	0.311	12.12	0.339	6.61
	CNP	0.330	10.78	0.338	6.53
	LAPR	0.337	10.45	0.342	6.46
CRF	LAPR+LAS	0.342	10.49	0.349	6.46
	LAPR+CNP	0.339	10.59	0.343	6.46
	ALL	0.344	10.59	0.349	6.40
DT	ALL	0.324	11.28	0.335	6.50

Table 7.11 *Confidence estimation results for the P2-cn configuration, showing UMAD(%) and NCE scores on dev10d (WER=29.9%) and eval09ns (WER=12.6%). DT= decision tree baseline.*

		dev10d		eval09ns	
Sys	Feat	NCE	UMAD	NCE	UMAD
—	CNP	−0.218	19.14	0.084	7.72
DT	CNP	0.223	14.01	0.315	6.75
	CNP	0.247	12.78	0.318	6.52
	LAPR	0.325	10.64	0.352	6.04
CRF	LAPR+LAS	0.328	10.61	0.357	6.01
	LAPR+CNP	0.351	10.35	0.397	5.67
	ALL	0.353	10.30	0.401	5.65
DT	ALL	0.297	11.79	0.342	6.24

Table 7.12 *Confidence estimation results for the P3-cn configuration, showing UMAD(%) and NCE scores on dev10d (WER=27.3%) and eval09ns (WER=11.1%). DT= decision tree baseline.*

For both sets of hypotheses, the CRF system which makes of the confusion network posteriors

[3]The decision trees (DTs) for the complete feature set do not employ a piecewise-linear mapping over a fixed number of intervals, as there is no straightforward way to do so for multiple feature dimensions.

(CNPs) as the sole predictor feature performs roughly as well as the decision tree-based system on the eval09ns dataset, and shows improved performance on dev10d, particularly in terms of UMAD. This result highlights the improvements that can be achieved through the use of the sequential modelling approach to confidence estimation taken here.

It is interesting to note that the LAPR feature yields improved performance over the CNP predictor feature in all systems. This implies that the methodology presented in this work for extracting the LAPR predictor feature is sound, producing posteriors that are effectively more accurate than the CNP posteriors. Thus, their use as a predictor feature in this work and in general is supported. The fact that greater improvements are seen in combining the CNP and LAPR features in the P3-cn system than in the P2-cn system highlights the difference between these features in this case, as they are calculated from different lattices. The inclusion of the Levenshtein alignment (LA) feature in the complete feature set does not result in a large performance improvement. This may be a result of the relative similarity between the 1-best and alternative hypotheses considered. The results on the complete feature set with decision trees show the utility of the predictor features in general, as the decision tree is also able to yielded improve confidence estimates. However, the large gap in performance between the decision tree and CRF-based approaches using these features (19% and 17% in NCE on dev10d and eval09 respectively), highlights the extent to which improvements are possible using the principled CRF-based approach of this work.

7.5 Summary Discussion

The details and results of a principled, flexible and elegant approach to word-level confidence estimation for ASR hypotheses based on CRF models are documented in this chapter. As a first proof of concept, experiments are presented which show that this approach is able to outperform a strong baseline when using a single predictor feature. The true power of this approach is however in its ability to effectively combine multiple predictor features (which may come from different information sources). The evidence encoded in the predictor features extracted from each such source is exploited by the model to yield improvements in the accuracy of confidence scores. A number of predictor features which proved to be of particular use for this task are also proposed, with the procedure in which

these may be calculated being detailed. The "hypothesis injection" technique was introduced as a means of obtaining predictor features for what may essentially be any reasonable hypothesis, from a corresponding ASR recognition lattice. The flexible definition of the confidence estimation task, such that it can be carried out for essentially any ASR hypothesis, is exploited further in the keyterm confidence work considered later in Chapter 9 of this thesis.

Systems in which various predictor features are combined yield substantial improvements over both the baseline, and single-feature systems. These improvements are evaluated in terms of word-level performance (with the NCE), and in terms of performance over sequences of words (utterances), using the UMAD metric put forward in this work. Analysis of DET curves for these systems also showed that performance improvements are achieved across a full range of operating points. Significance tests applied to the various CRF-based systems served to give additional weight to the results, proving that the gains observed by the relevant CRF-based systems are not by chance.

Experiments showed that in general, the more different the source of a predictor feature is from the baseline LAPR feature, the more use it has in combination with this feature. This was particularly true in the case of the decoupled posteriors, scores from an alternative sub-word recogniser, and features extracted from different lattices to those from which the underlying hypothesis is generated.

The details of extensions to the "core" set of feature functions were presented. Of these, the spline feature functions proved to be a particularly useful addition to the set of feature functions used by the models.

The novel application of hidden-state CRF models to confidence estimation is proposed and investigated further. The improvements achieved by models of this type are shown to in fact be larger than that achieved when including additional predictor features in the linear-chain CRF. These improvements in performance are attributed to the ability of the hidden-state models to capture a hidden dynamic transition structure, related to the notion of regions or pools of confidence. Modelling this higher-order structure implicitly through hidden variables ultimately has a positive impact on the model, and consequently helps to improve the accuracy of the confidence scores estimated.

Chapter 8

Sub-word-level Confidence Estimation

In Chapter 7, the task of estimating confidence scores for words in ASR hypotheses was investigated. In this chapter, the task of confidence estimation is defined at a finer level of granularity than the word level, this being the sub-word level. The task is therefore redefined as that of estimating confidence scores which indicate whether or not a particular sub-word is believed to be truly correct within an overall, word-level ASR hypothesis.

In the CRF-based framework for confidence estimation (CE) proposed and developed in this work, this implies that the task of the CRF model becomes that of predicting a sequence of `Correct` and `Incorrect` labels for sub-word hypotheses. The entire model therefore operates on the sub-word level, instead of the word level. The stream of input predictor features and corresponding labels in the data are therefore related to sub-word hypotheses.

Modern large vocabulary speech recognisers make use of sub-word level representations for acoustic modelling (e.g. Lee *et al.* (1993); Odell (1995); Gales and Young (2007); Livescu *et al.* (2012)). This implies there is already a significant amount of information defined on the sub-word level, which is to some extent readily available for use in confidence estimation. Word-level information is also expected to be useful on the sub-word level, and should not be disregarded. Such information is expected to be particularly useful in situations where the output of the sub-word-level model is to be mapped to yield confidence scores on the word level. It is also likely that the degree of confidence which should be placed in individual sub-words is closely related to that of the words they constitute. Modelling the confidence estimation problem at the sub-word level makes it possible for systems of this type to utilise

information from both the word and sub-word level. The development of a confidence estimation system which is capable of exploiting information from both levels to best effect is therefore of interest. The CRF-based framework proposed and refined as part of this work is investigated as a means of achieving this.

The sub-word-level predictor features utilised for the confidence estimation task at this level, and the sources from which they are extracted, are presented in the following two sections. Thereafter, the details of feature functions engineered with the aim of capturing specific aspects of the sub-word-level problem are provided. Based on the resulting framework for sub-word-level confidence estimation, the outcomes of a number of experiments are presented. The utility of various combinations of predictor features, applied within different overall approaches to sub-word-level confidence estimation are presented separately and subsequently discussed.

A number of the contributions detailed in this chapter are based on published work detailing a CRF-based approach to the task of sub-word-level confidence estimation, and the application of hidden-state CRFs to the CE task in general (Seigel and Woodland 2012).

8.1 An Alternative Sub-word Recogniser

Word-level confidence estimation experiments showed that predictor features which are based solely on the acoustic or language model scores were some of the most useful in improving the accuracy of the confidence measures (see Section 7.4.3). This fact is made evident through the gains achieved when these predictor features are combined with the lattice arc posterior ratio (LAPR), to which both the acoustic and language model contribute. It is understood that these "decoupled" predictor features effectively allow the confidence estimation model to capture information pertaining to the degree to which a particular hypothesis matches with the acoustics or the language in isolation, irrespective of the other model. However, acoustic model scores decoupled in this way are still not truly independent of the language model. This is due to the fact that, to some extent, the language model constrains the hypothesis space of the lattice over which these scores are computed. Hypotheses which match the acoustics closely may still be pruned away from the search space if they are not possible, or likely, according to the language model.

Based on this motivation, an alternative sub-word-level acoustic model (i.e. a sub-word classifier) is used in this work. This model serves to provide acoustic predictor features which are completely decoupled from the language model. It also represents a significantly different source of information in a more general sense. This is a result of the fact that the architecture, training procedure and other aspects of this recogniser are quite different from that employed by the underlying system.

The goal of the alternative recogniser is to provide competing information through predictor features, rather than competing hypotheses (as would be the case in system combination). This recogniser need not, and should therefore not, require an inordinate amount of additional design effort. A relatively lightweight frame-level recogniser is therefore utilised. This recogniser is based on a neural network or multi layer perceptron (MLP), which predicts grapheme targets for each frame of audio. Graphemes are the representation used by this recogniser as it is the sub-word representation employed by the underlying recogniser in this work. The MLP recogniser is trained on 140 hours of Arabic broadcast news data, the GALE p4r3 dataset (Olive *et al.* 2011), using the ICSI `QuickNet` MLP neural network software (Johnson 2004). For each frame of the audio, 14 perceptual linear prediction (PLP) features are extracted, with deltas and double-deltas being computed in addition. The complete input feature vector for this MLP consists of the features for the current frame, as well as those for four context frames to the left and right of the current frame, thus resulting in a window size of 9 frames. The MLP structure includes one hidden layer consisting of 3500 nodes, and an output layer comprising 37 softmax nodes (one for each grapheme target). The cross validation accuracy that is achieved by this recogniser on this data is 66.6%.

8.2 Sub-word-level Predictor Features

Sub-word-level predictor features are extracted from two sources. The first of these are the output lattices of the underlying recogniser for which confidence estimation is being performed. The second is the alternative recogniser, the details of which are provided in Section 8.1.

8.2.1 *A Lattice-based Predictor Feature*

The underlying recogniser is an obvious source of sub-word-level information, as the acoustic models used are defined on the sub-word level (graphemes). The word-level lattices output by the system are marked up with the timing information for individual sub-word units which comprise the word arcs. Given this information, predictor features can be calculated directly from the lattice, as a "local hypothesis space" of competing sub-words for a hypothesised sub-word can be determined. The procedure for doing so is in many ways similar to that described for the word-level task in Section 6.1.1. The only significant difference between the technique as used here and that depicted in Figure 6.1 on page 71, is that the arcs in the lattice over which features are computed are sub-word arcs rather than word arcs.

On the word-level, posterior probabilities for hypothesised words are calculated directly from the lattice, resulting in the lattice arc posterior ration (LAPR) predictor features. A posterior probability defined over sub-word units may be computed in a similar manner. Arc posterior probabilities $p(a|\mathbf{O})$ are calculated for each of the word-level arcs a in a lattice, conditioned on the acoustic observation vectors $\mathbf{O}$ (as described in Evermann and Woodland (2000*b*)). For each sub-word S, an approximation is made for the sub-word arc posterior $p(S|\mathbf{O})$, such that these are assumed to be equal to the word-level arc posterior of the word to which a sub-word belongs. Each sub-word in a given word will therefore have the same sub-word arc posterior score. This approximation is necessary due to the fact that the lattices used are defined on the word-level, with word-level acoustic and language model scores being available. Using the sub-word timing information, the word-level arcs are however decomposed into a representative set of sub-word-level arcs with the correct timing. This makes it possible for predictor features to be calculated over an appropriate set of sub-word arcs $\mathcal{I}$, which intersect with a hypothesised sub-word in the lattice. For a hypothesised sub-word G_i at position i along the 1-Best path $\mathbf{G}$ through a lattice, the posterior probabilities of sub-word arcs S in the set of intersecting arcs $\mathcal{I}$ with the same sub-word(S) as the hypothesised sub-word G_i are summed. This sum is normalised by the sum of the posterior probabilities for all sub-word arcs in $\mathcal{I}$, yielding the following expression for the lattice sub-arc posterior ratio (LSAPR) :

$$\text{LSAPR}(G_i, \mathcal{I}) = \frac{\sum_{S \in \mathcal{I}} \delta(\text{sub-word}(S), G_i) p(S|\mathbf{O})}{\sum_{S \in \mathcal{I}} p(S|\mathbf{O})}. \tag{8.1}$$

It should be noted that the LSAPR predictor feature has been defined here for sub-words within a

1-Best sequence, but this may easily be extended to other types of hypotheses.

8.2.2 *Predictor Features from the Alternative Recogniser*

The other source of sub-word level predictor features considered in this work is a separate or alternative recogniser. The output of this system must be related to the hypotheses generated by the underlying recogniser in some way, such that useful predictor features may be extracted from it. This is not entirely straightforward, as the MLP recogniser performs frame-level classification. However, the smallest representable unit in the output of the underlying recogniser is a complete sub-word (grapheme), which may span multiple frames. In order to use this information effectively in generating predictor features, and to streamline the overall feature extraction process, the recognition lattice is augmented with the scores output for each frame before computing the complete set of predictor features. Given a particular sub-word arc in the lattice, features may be computed based on the alternative recogniser output over the required span of frames. The definition of various predictor features based on the scores from the alternative recogniser will be provided in this section.

The purpose of the alternative recogniser is to provide complementary information for the hypotheses present in the ASR system lattices. Computing a posterior probability similar to that calculated directly from the lattice for a given hypothesised sub-word is therefore useful. Given a particular sub-word G_i which is hypothesised to have occurred along the 1-Best path through the lattice between the frames t_b and t_e, the alternative sub-word posterior (ASWP) may be expressed as follows:

$$\mathrm{ASWP}(G_i, t_b, t_e) = \frac{\sum_{t\in[t_b,t_e]} \sum_{S\in\mathcal{S}} \delta(\text{sub-word}(S), G_i) p(S|t)}{\sum_{t\in[t_b,t_e]} \sum_{S\in\mathcal{S}} p(S|t)} \tag{8.2}$$

where t is a particular frame, S is one of the sub-word targets (a grapheme) in the complete set of graphemes $\mathcal{S}$. The Kronecker delta function (δ) matches the sub-word identities sub-word(S) and G_i, and $p(S|t)$ is the posterior probability output by the alternative recogniser at time t corresponding to the sub-word S. The ASWP posterior provides information relating to the alternative recogniser's belief that the sub-word hypothesised by the underlying system is in fact present in the audio. There is no language model contribution within this score, such that it is based primarily on the acoustic evidence and the prior probability of sub-word classes.

The grapheme which is hypothesised by the alternative recogniser as being the most probable over a given time interval is also of interest. This information can be contrasted directly with the hypothesis of the underlying recogniser. The most probable grapheme is assumed to be that which has the highest posterior probability at any time during the span from t_b to t_e. This sub-word hypothesis will be referred to as the best alternative sub word (BASW), and is expressed as follows:

$$\text{BASW}(t_b, t_e) = \arg\max_{S \in \mathcal{S}} \max_{t \in [t_b, t_e]} p(S|t) \tag{8.3}$$

where the arg max operation selects the sub-word unit with the highest posterior score calculated over all frames t in the interval of interest using the max operation. The identity of this sub-word is in itself useful. It is also possible to use this as the reference sub-word (G_i) in equation 8.2 to yield the best alternative sub-word posterior (BASWP), defined as follows:

$$\text{BASWP}(t_b, t_e) = \text{ASWP}(\text{BASW}, t_b, t_e). \tag{8.4}$$

The MLP-based alternative recogniser provides alternative scores for sub-words in the lattice. However, these scores may not always be as accurate as those already represented by the lattice. This is due to the fact that that the alternative recogniser is not as powerful as the acoustic model employed by the underlying recogniser. It is therefore useful to have a means for quantifying the degree of certainty that the alternative recogniser has in the scores it generates over a range of consecutive frames. A predictor feature based on the entropy in the posterior probabilities output by the MLP recogniser, which is similar to the per-frame entropies used as a confidence measure in Williams and Renals (1999) or that used for out-of-language detection in Motlícek (2009), is therefore proposed as a manner whereby this may be represented. The entropy in the alternative system (HA) is calculated for the output probabilities $p(S|t)$ generated by the alternative recogniser. This is averaged over the t frames corresponding to the reference sub-word hypothesised by the underlying system as follows:

$$\text{HA}(t_b, t_e) = -\frac{1}{t_e - t_b} \sum_{t \in [t_b, t_e]} \sum_{S \in \mathcal{S}} p(S|t) \log(p(S|t)). \tag{8.5}$$

8.3 Feature Engineering

The "core" set of feature functions presented in Section 6.2 are used by the models applied to the sub-word-level task. However, a number of additional feature functions are developed to augment the

CRF framework, and thereby enhance its suitability for this specific task. These feature functions are detailed in the sections which follow.

8.3.1 *Word Boundary Feature Functions*

Carrying out the modelling task at the sub-word level implies that the word-level transition structure is no longer represented explicitly in the model. This is not necessarily a concern when the task is that of estimating sub-word confidence scores. However, if the sub-word model should be capable of producing accurate word-level confidence measures (as in the word-targeted sub-word task of Section 8.4.3), the model must capture this information. The location of clear transitions between words, and the effect this has on the relation between the predictor features and the output label, is determined experimentally in this work to be of paramount importance for accurate measures to be produced at the word level.

A long-range transition feature function, similar to the skip-chain feature functions introduced in Sutton and McCallum (2004), was therefore implemented to compensate for this loss of higher-level transition structure. This feature function is only active on word boundaries. Boundaries are marked explicitly in the input vector of predictor features, and are passed as an argument to the feature function. In this way, the word-level transitions are modelled separately from the sub-word-level transitions. The resulting definition of the word boundary feature function (WB) is the following:

$$\mathrm{WB}_{y',y,h',h}(\mathbf{X}[b]_i) = \delta(\mathbf{X}[b]_i, \texttt{``Word Start''})\, t_{y',y,h',h} \tag{8.6}$$

where $\mathbf{X}[b]_i$ is the predictor feature that indicates whether the observation is at a word boundary (`''Word Start''`) or within a word (`''Inside Word''`), the Kronecker delta δ is used to match this feature to the template value of a word boundary (`''Word Start''`), and $t_{y',y,h',h}$ is the generic transition feature function (defined in equation 6.10).

8.3.2 *String Match Feature Functions*

Within the sub-word confidence estimation framework, the identity of the sub-word which is hypothesised to have occurred by the underlying recogniser is known. In addition, the identity of the sub-word assigned the highest score by the alternative recogniser on the same interval as the hypothesised

sub-word (BASW) may be determined. These two pieces of information are potentially useful when considered in relation to one another. Discrepancies between them may indicate situations where the 1-Best hypothesis doesn't match the acoustic evidence. In order to investigate the use of this information, feature functions which are used to act directly on discrete observations such as the sub-word identities, and the scores corresponding to them, are proposed in this work. These feature functions are discrete in nature, and are effectively extended versions of the core discrete feature function presented in Section 6.2.1. The primary difference being that these are defined such that they act on more than one discrete dimension of the same (current) predictor feature vector.

One set of feature functions of this type are a pair of symmetric discrete "string match" feature functions (returning 1 or 0). These feature functions (referred to as SM), are binary in nature and compare the discrete string values of two specific elements in the predictor feature vector. One of these predictor feature is active if the strings are matched, and the other if they are different. Another pair of feature functions which extend upon the definition of the SM feature functions are also proposed. These return a continuous value instead of the discrete indicator variable. This pair of feature functions (referred to as SMV), therefore perform the same string match function as SM, and return the value of an arbitrary continuous predictor feature if these strings are either matched or different. The SM and SMV feature functions are expressed as follows:

$$\mathrm{SM}_{y,h}(\mathbf{X}[s_1]_i, \mathbf{X}[s_2]_i, Y_i, H_i) = \delta(\mathbf{X}[s_1]_i, \mathbf{X}[s_2]_i)\ \delta(Y_i, y)\ \delta(H_i, h) \tag{8.7}$$

$$\mathrm{SMV}_{y,h}(\mathbf{X}[s_1]_i, \mathbf{X}[s_2]_i, \mathbf{X}[p]_i, Y_i, H_i) = \mathbf{X}[p]_i\ \delta(\mathbf{X}[s_1]_i, \mathbf{X}[s_2]_i)\ \delta(Y_i, y)\ \delta(H_i, h) \tag{8.8}$$

where the Kronecker delta is used to match the current label Y_i to the label value y for which the feature function is defined, as well as matching the current hidden state value H_i to the template hidden state value h. The variables $\mathbf{X}[s_1]_i$ and $\mathbf{X}[s_2]_i$ are the strings to be compared, and $\mathbf{X}[p]_i$ is the continuous predictor feature for which the first order moment parameters will effectively be estimated (in the case of the SMV feature function). The feature functions in this section have been expressed in their most general form and therefore include hidden variables. However, these definitions hold for the standard linear-chain CRF model, in which case the number of possible hidden state values is 1.

8.4 Experiments

Two separate tasks which are within the scope of sub-word-level confidence estimation are investigated in this work. These tasks are the following:

1. Improving the accuracy of confidence scores assigned to **sub-word** units by combining information from the word and sub-word level. This will be referred to as *direct sub-word confidence estimation*. All aspects of this task are defined at the sub-word level.

2. Improving the accuracy of confidence scores assigned to **words** by combining information from the word and sub-word level. This will be referred to as *word-targeted sub-word CE*. Here, the models used for this task act on the sub-word level, but word-level confidence estimates are required.

Other than aspects of the models themselves, the defining difference between these two approaches is effectively in the final output they produce. This may either be scores relating to a sequence of sub-words, or a sequence of words for the direct and word-targeted tasks respectively.

Experiments exploring aspects of the two sub-word-level confidence estimation tasks are presented in the sections which follow. The experimental setup in terms of the datasets used, and the underlying ASR systems for which confidence estimation is carried out, are the same as those used in the word-level confidence estimation experiments. The details of this setup are provided in Section 7.2. One notable difference is that the recogniser output lattices are marked up with sub-word-level timing information. This makes it possible for the predictor features detailed in Section 8.2 to be computed. Unless otherwise specified, all continuous predictor features are represented in the CRF models using the standard spline configuration, which makes use of 8 evenly spaced intervals.

8.4.1 Direct Sub-word-level Confidence Modelling

Sub-word level predictor features may be utilised to estimate the degree to which each individual sub-word within a sequence of sub-words is believed to be correct. The confidence estimation task defined in this manner is a natural extension of the approach taken in the word-level task onto this level. The sequential nature of the CRF modelling approach taken in this work is particularly suited to this task.

These models are capable of capturing the dynamic nature of runs of consecutive correct or incorrect ASR hypotheses. Here, this sequence structure may be represented at a finer level of granularity than is possible in the word-level confidence estimation systems. Another benefit of this approach, is that in carrying out the modelling task at the sub-word level, the predictor features defined on this level are able to have a significant local impact on the confidence scores for the sub-words with which they are related. The aforementioned factors result in a principled modelling approach for estimating accurate sub-word-level confidence scores. The task is formulated as one in which the CRF models are effectively required to perform binary classification of sub-words as either being correct or incorrect, and the marginal probability associated with the `Correct` label is used as a confidence score.

One particular application in which accurate sub-word confidence scores are useful, is that of spoken term detection. This application is in fact explored in Chapter 9 of this work. Many approaches to spoken key term detection make use of a sub-word-level approach in which sub-word units are concatenated together (for a discussion of these approaches see Section 5.1.3). The motivation for this sub-word-level search is that in ensures a large number of keyterm hypotheses are generated, while also making it possible for out-of-vocabulary (OOV) keyterms to be hypothesised. Improving the confidence scores for the constituent sub-words of keyterm detections will result in more accurate overall scores for the keyterm hypotheses. Another potential application of sub-word-level confidence scores is in unsupervised speaker adaptation. Techniques have been proposed in which confidence thresholds are applied to automatically transcribed recogniser output. Here, only those acoustic models corresponding to words which are selected as being likely to have been transcribed correctly are adapted. The same principle may essentially be applied to individual sub-words and their confidence scores. Carrying out data selection at the sub-word level will ultimately result in more data being retained, thus making greater use of the available information. In addition, having more accurate scores to which these thresholds are applied will result in more accurate distinctions being made between data which should and shouldn't be considered for adaptation/data selection.

In mapping the word-level configuration for confidence estimation to the direct sub-word confidence estimation task, the target labels are naturally different. Here, these labels are sub-word `Correct` or `Incorrect` labels. The standard NIST `SCLite` scoring tool used in this work evaluates word-level performance. Sub-word-level target labels must however be obtained. This is achieved by carrying

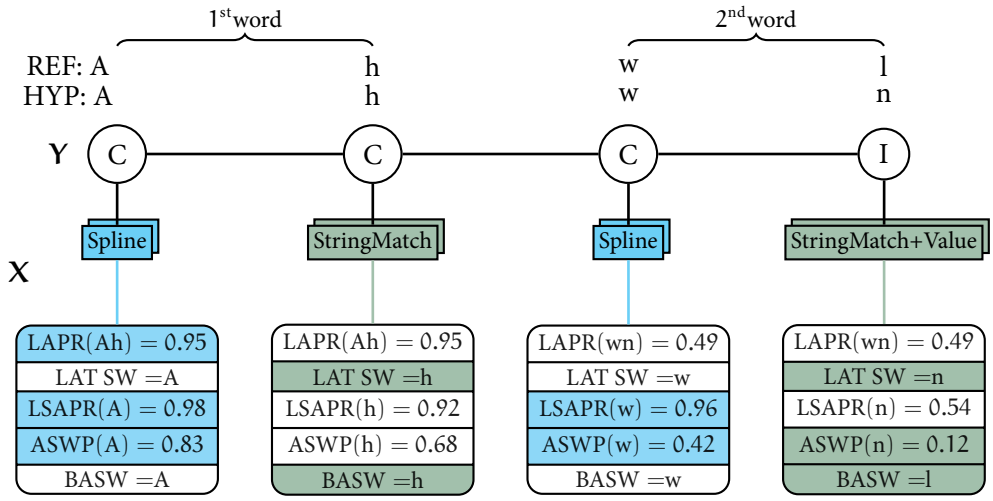


Figure 8.1 *Figure illustrating direct sub-word-level confidence estimation using CRF models, with feature functions shown acting on appropriate dimensions of the feature vector.*

out the normal word-level scoring and alignment process, such that words in the ASR output can initially be labelled as either being correct, or corresponding to an insertion or substitution error. For words which are inserted, each grapheme (sub-word) within the hypothesised word is labelled as being incorrect. However, for substituted words, there may be some overlap between the hypothesised and correct word in the reference. This implies some sub-words within the word may have effectively been hypothesised correctly, and the data should reflect this situation. The scoring process outputs the correct aligned word for substitutions. A Levenshtein alignment is therefore carried out between the reference and hypothesised grapheme sequences. After alignment, this yields appropriate `Correct` or `Incorrect` labels for the graphemes hypothesised by the ASR system.

An illustration of the approach taken for direct sub-word-level confidence estimation is shown in Figure 8.1. As has already been mentioned, the underlying recogniser makes use of graphemes as the sub-word representation. The example illustrated in this figure is that of estimating confidence scores for the sequence of graphemes `A-h w-n` (two words consisting of two graphemes each). The reference transcription corresponding to this hypothesis is `A-h w-l`. This implies that there is one misrecognised grapheme in the second word. The sequence of labels indicating whether the sub-word is correct or incorrect (abbreviated to `C` or `I`), which are assigned by the CRF is depicted as $\mathbf{Y}$. The feature vector of combined word and sub-word level predictor features is $\mathbf{X}$. At each point in the sequence, a par-

ticular CRF feature function (`Spline`, `StringMatch`, or `StringMatch+Value`) is highlighted. This serves to illustrate the dimensions of the feature vector (and variable types) that these feature functions typically act on. It should however be noted that all feature functions will act on all relevant dimensions of the feature vector at each point in the sequence. The `Spline` feature functions are generally used to represent all continuous predictor features, in which the default setting of 8 evenly spaced intervals is applied. The `StringMatch` and `StringMatch+Value` feature functions both match the identities of the 1-Best grapheme hypothesised by the ASR system, with the highest scoring grapheme output on the same interval by the alternative MLP recogniser. The SM feature functions are discrete feature functions, returning 1 or 0 if the aforementioned identities are matched. The SMV feature functions are continuous in nature, and are related to the continuous score output by the alternative recogniser for the hypothesised grapheme (ASWP).

The CRF model classifies each observation within a sequence as belonging to a particular class (`Correct` or `Incorrect`). It does not estimate a continuous value for each such observation. However, as was carried out in the word-level confidence estimation systems of this work, the confidence score for an individual sub-word is computed as the marginal probability assigned by the model to the label `Correct` for every observation within the sub-word sequence. This marginal probability is computed by running the forward-backward algorithm as part of the decoding phase for the CRF during test.

In order to evaluate these systems, the word-level normalised cross entropy (NCE) metric is modified such that it is calculated over sub-words instead of words. The resulting normalised sub-word cross entropy (NSWCE) is therefore defined as follows:

$$\text{NSWCE} = \frac{\left(H_{max} + \sum_{S \in \mathcal{C}} \log_2(\hat{P}(S)) + \sum_{S \in \mathcal{I}} \log_2(1 - \hat{P}(S))\right)}{H_{max}} \tag{8.9}$$

where $\hat{P}(S)$ is the confidence score for a particular sub-word S, $\mathcal{C}$ is the set of correct sub-words and $\mathcal{I}$ is the set of all incorrect sub-words. Given n correct sub-words in the hypotheses out of N total sub-words, the empirical average probability of any sub-word being correct is estimated as being $P_c = \frac{n}{N}$, resulting in the usual form of the maximum entropy in the confidence scores:

$$H_{max} = -n \log_2(P_c) - (N - n) \log_2(1 - P_c). \tag{8.10}$$

8.4.1.1 Results

The results of experiments in which a CRF-based approach is taken to perform direct sub-word-level confidence estimation are presented in Table 8.1 on the following page. A number is assigned to each system shown in the table to aid comparisons in the discussion which follows. The results of statistical significance tests carried out for particular systems which provide interesting contrasts are also shown.

As in the case of the word-level confidence estimation experiments, the baseline system considered for the sub-word-level confidence estimation experiments utilises a decision-tree-based segmentation of a predictor feature's range, with a subsequent piecewise-linear mapping applied over each such interval. However, it was necessary to re-train the parameters of these decision trees for the new task, as the target `Correct/Incorrect` labels are related to graphemes for these experiments, as opposed to words. Two baselines are shown in Table 8.1 on the next page, one which is based on the word-level LAPR predictor feature (2), and one which is based on the sub-word posterior extracted from the lattice (3). Different decision trees were trained for each of these systems. It is interesting to note that the word-level LAPR predictor feature is still relevant, and even outperforms the sub-word level LSAPR posterior on dev10d. This result may be attributed to the fact that although the task is defined at the sub-word level, the hypotheses for which confidence scores are ultimately being estimated are actually sub-words within word-level hypotheses. It should be noted that the empirical accuracy P_c used in computing the NSWCE (equation 8.10), is significantly different from the word accuracy in the word-level confidence estimation system. The sub-word-level accuracy is higher than the word-level accuracy, at 85.4 for dev10d (71.4% word accuracy) and 95.3% for eval09ns (87.5% word accuracy). This change in error rate profile has a direct effect on the evaluation metrics used, which will fall into different ranges from those previously reported for the word-level experiments.

As another point of comparison, a standard word-level confidence estimation model based on the LAPR predictor feature was evaluated on the sub-word level (system 1). The output of this system was converted to a sub-word-level representation by assuming a uniform distribution of sub-word scores within a word. Under this assumption, the sub-word-level scores are approximated as being equal to the word-level scores. These word-level scores are therefore repeated over each sub-word unit which comprises a given word. The mismatch between the word and sub-word level tasks clearly has a significant impact on performance in this approach, as made evident through the poor NSWCE perform-

System		dev10d		eval09ns	
Num	Description	NSWCE	UMAD	NSWCE	UMAD
①	WCRF: LAPR	0.092	10.43	-0.029	11.38
②	SWDT: Baseline (LAPR)	0.208	13.96	0.266	6.88
③	SWDT: Baseline (LSAPR)	0.162	14.16	0.265	6.79
④	SWMaxEnt: LAPR	0.201	14.08	0.260	6.91
⑤	SWCRF: ASWP	0.076	16.36	0.091	8.33
⑥	SWCRF: LSAPR	0.210	12.18	0.263	6.58
⑦	SWCRF: LAPR	0.215	13.00	0.256	6.69
⑧	SWCRF: LAPR 9A	0.220	12.93	0.262	6.69
⑨	SWCRF: LSAPR+ASWP	0.221	12.21	0.266	6.62
⑩	SWCRF: LAPR+BASWP	0.237	12.33	0.269	6.54
⑪	SWCRF: LAPR+LSAPR	0.251	11.73	0.305	6.22
⑫	SWCRF: LAPR+ASWP	0.254	12.15	0.284	6.45
⑬	SWCRF: LAPR+BASWP+ASWP	0.254	12.17	0.285	6.46
⑭	SWCRF: LAPR+ASWP+LSAPR	0.263	11.68	0.310	6.21
⑮	SWCRF: ⑭ +SMV(BASWP)	0.265	11.66	0.313	6.19
⑯	SWCRF: ⑭ +SM	0.265	11.70	0.315	6.13
⑰	SWCRF: ⑭ +WLEV	0.274	11.63	0.320	6.25

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
System ⑦	Baseline	0.002	1	45.8
System ⑭	Baseline	0.049	0.001	85.8
System ⑰	Baseline	0.060	0.001	87.9

Table 8.1 *Results and evaluation of various CRF-based systems which operate solely on the grapheme level, to estimate grapheme confidence scores. Models are evaluated in terms of NSWCE and UMAD on dev10d and eval09ns. All continuous predictor features are represented using a spline approximation, over 8 evenly-spaced intervals, unless indicated otherwise. SWCRF=sub-word-level CRF model, WCRF=word-level CRF model, SWDT=sub-word-level decision tree, SWMaxEnt=sub-word-level maximum entropy model. Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNSWCE = average difference between NSWCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

ance scores achieved by this system. These scores are considerably lower than the baseline. The reason this model is unable to yield competitive performance is due to the mismatch in the level at which the

tasks are defined in training and evaluation. The word-level system has no means of distinguishing between correct sub-words within words which are correct, and those which are within words which are incorrect. In a sense, the word-level system is constrained through the implied assertion that all sub-words within a word must have the same label. This is most certainly not true when considering the problem of classifying individual sub-words as being correct or incorrect. This result serves to prove that should sub-word-level confidence scores be required, a word-level confidence estimation system defined entirely at this level should certainly not be used to obtain estimates for these scores.

A sub-word-level CRF system which makes use of the LAPR predictor feature in isolation (7) is shown to achieve marginally improved NSWCE performance over the baseline on dev10d, but yields worse performance on eval09ns. This result is however reversed when considering the UMAD metric. This may be attributed to the fact that the average length of utterances in the dev10d dataset is shorter than in eval09ns. On the word-level, the differences in the average number of words in each utterance for the datasets is 3. The corresponding difference on the sub-word level is an average of 17 graphemes. The margin is therefore greater in the grapheme-level task, and the sequential nature of the CRF may result in biases towards longer sequences. Furthermore, the training data is more closely matched to the eval09ns data, where the difference in the average number of graphemes between these two datasets being only 2.5. On the other hand, if the LSAPR predictor feature is used as the sole predictor feature in the CRF (6), the NSWCE performance over both datasets is similar to that of the LAPR baseline, and improved by a fair margin over the LSAPR baseline. This difference in performance of each predictor feature used in NSWCE (particularly on eval09ns), suggests that these effectively embody quite different types of information. The result obtained for the combination of these two predictor features further supports this conclusion, as large gains are obtained in both the NSWCE and UMAD metrics (system 11). The results for both metrics outstrip the best performance figures obtained in systems where these features are used in isolation.

A significance test is carried out to determine whether the results for the single feature CRF (system 7) are significant compared to the baseline. These results are shown to not be statistically significant, with the p-value for the test only being less than 1. This system yields an average improvement in NSWCE over the evaluation datasets of only 0.002, which is in line with the fact that the test shows there is no significant difference between the results. These systems also make use of only a single

predictor feature (LAPR), which is not actually defined on the sub-word-level at which these systems operate. The CRF system uses this feature in isolation, and this feature is static to a large degree, as it is repeated over consecutive sub-words within a word. It may therefore be expected that the CRF is not able to leverage the sequence information in consecutive words to a large extent. This will also result in the CRF system being more similar to the non-sequential baseline. In addition, it is seen that the CRF-based system (7) is able to outperform the baseline on just below half of the snippets at 45.8% (the SIP ratio). The substantive analysis in terms of low SIP and NSWCE performance, coupled with the significance test results, show that the systems are indeed quite similar, with the CRF system being slightly worse than the baseline in some cases.

Two of the grapheme-level scores extracted from the alternative recogniser are the posterior of the grapheme hypothesised by the underlying recogniser (ASWP), and the grapheme posterior with the highest score (BASWP). These predictor features contribute in a two-way combination with the LAPR predictor feature, with ASWP yielding the largest gains (system 9). However, including the BASWP in a three-way combination with LAPR and ASWP yields no substantial gain. This outcome is likely a result of two factors. The first is that the ASWP and BASWP scores are the same when the two recognisers agree on the sub-word hypothesis, and consequently represent the same information. The second factor is that when the alternative recogniser's highest scoring grapheme is different from the grapheme hypothesis in the lattice, the score for this alternate grapheme has no real relation to the confidence of the grapheme of interest - which is that hypothesised by the underlying recogniser.

The system which yielded the best performance using primarily sub-word-level predictor features is one in which the word-level LAPR predictor feature is combined with the sub-word-level LSAPR feature extracted from the lattice and the ASWP feature extracted from the alternative recogniser (system 14). The results of this system are also proven to be significant, with a p-value of less than 0.001. This system also outperforms the baseline in 85.8% of the audio snippets. Improvements over the LAPR-only sub-word-level configuration (system 7) of 22.3% and 21.1% relative in NSWCE on dev10d and eval09ns are achieved by this system. This result serves to highlight the fact that in performing sub-word-level confidence estimation, the use of suitable predictor features defined at the same level of granularity is of considerable importance.

Using the string match (SM) and string match with value (SMV) feature functions to match the 1-

Best grapheme and that hypothesised by the alternative recogniser (systems 16 and 15 respectively), result in slight performance improvements. The value of the best alternative sub-word posterior (BASWP) is used as the feature for which the mean is learnt through the SMV feature functions in system 15. It is clear that directly including the information in the agreement of the two sets of sub-word hypotheses through the use of these feature functions does not contribute a great deal to performance. This may be explained by the fact that information relating to the agreement between the two hypotheses is already effectively captured by the model through the posterior scores for these sub-words, which are also used by these systems.

Including the full set of word-level features (system 17) resulted in an incremental improvement in performance for this task, yielding the best performance achieved by models of this type. The relative improvement in NSWCE score over the baseline with this additional information is 31.7% and 20.3% on dev10d and eval09ns respectively. This result is shown to be statistically significant (with a p-value less than 0.001). The system is also capable of yielding improved NSWCE scores over the baseline in 87.9% of the snippets. Compared with system 14, this system achieves relative improvements of 4.2% and 3.2% in NSWCE on dev10d and eval09ns respectively. Whilst considerable, the scale of these improvements is less prominent than was observed when including sub-word-level features in the model. Including many additional word-level predictor features is shown to be useful. as they are able to contribute in improving sub-word level scores to some extent. However, the fact that these are naturally defined on the word level implies their impact is diminished from that observed for the word-level experiments.

An analysis of the errors that result when applying a threshold (0.5) to the combined output (i.e. dev10d and eval09ns datasets) for the best system (17), showed that this full-featured CRF system yields a relative reduction in error rate of 6.6% over the baseline (which has an error rate of 12.2%). This reduction is also 1.7% larger than that achieved by the CRF system using the LAPR and LSAPR features, which in turn yields improved performance over the single-feature LSAPR and LAPR systems.

8.4.1.2 Analysis of Detection Error Trade-off Curves

The performance of direct sub-word confidence estimation using CRF models is evaluated using DET curves to investigate performance of these systems over a wider range of operating points. The DET

curves for representative systems are shown in Figures 8.2 and 8.3 on the next page for the eval09ns and dev10d datasets respectively. Performance curves are plotted for the full-featured CRF, the baseline, and the single-feature CRF using the word and sub-word posteriors (LAPR and LSAPR). These results are analysed in the discussion which follows.

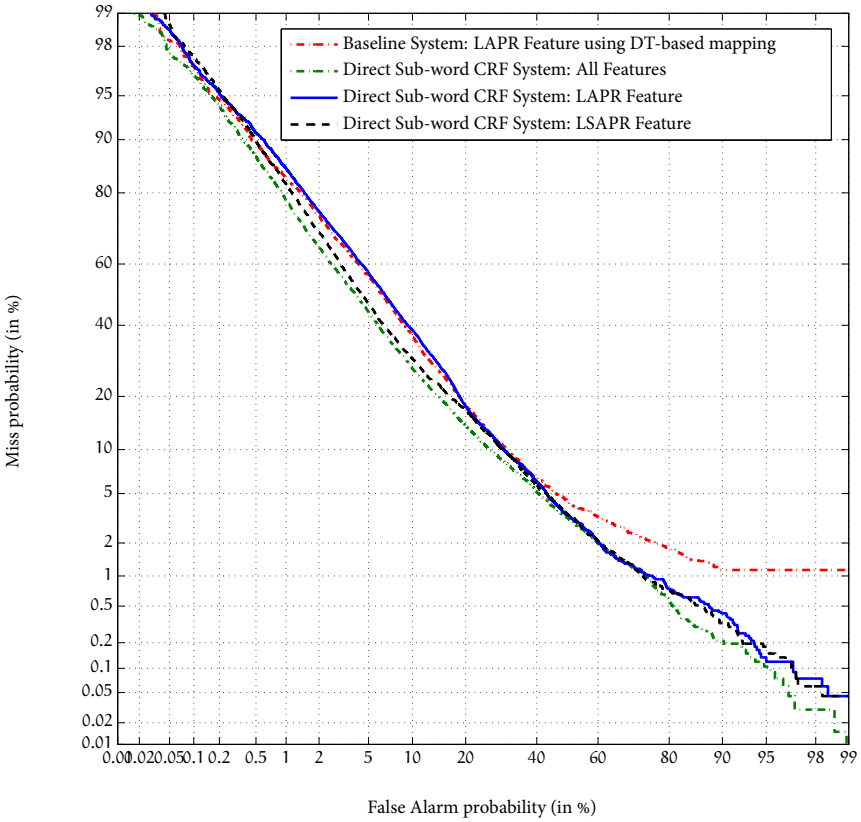


Figure 8.2 *DET curves showing the performance on eval09ns of CRF systems for direct sub-word-level confidence estimation. Curves for the baseline system, single-feature CRFs (LAPR and LSAPR) and full-featured CRFs are shown.*

The DET curve plots in Figures 8.2 and 8.3 on the next page show once again that, as in the word-level task, the CRF-based systems outperform the decision tree baseline. This is particularly true in the regions with higher false alarm probability, where the baseline performance reaches a plateau whilst the miss probability continues to decrease for CRF-based systems. The performance of systems using the LAPR and LSAPR predictor features in isolation show interesting trends. Firstly, the LAPR-only CRF system yields similar performance to the baseline for a wide range of operating points. However, above

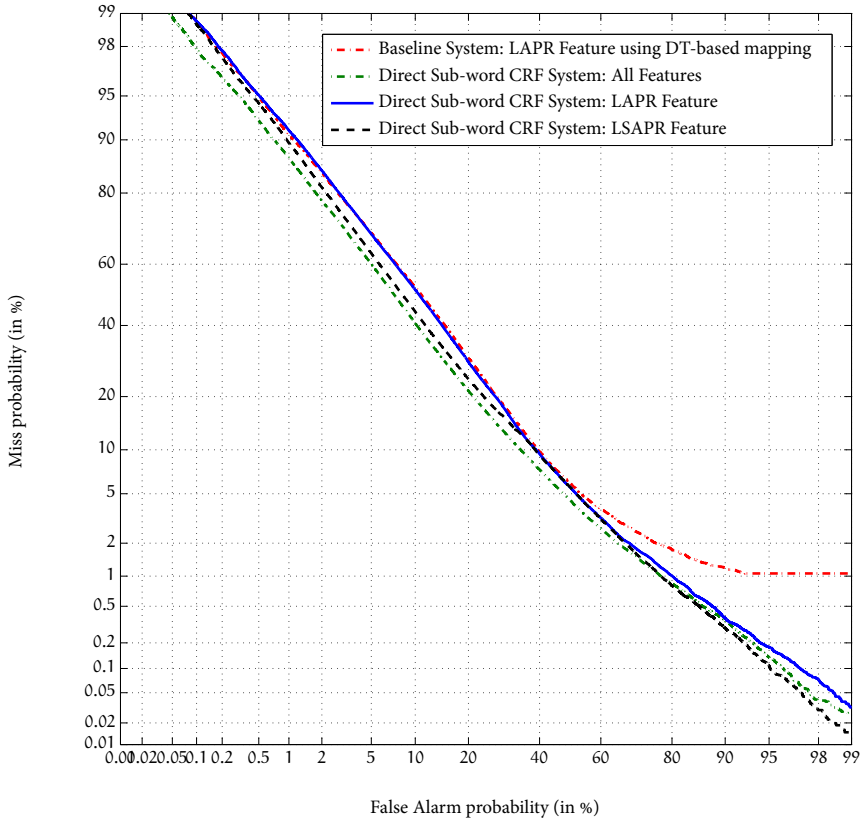


Figure 8.3 *DET curves showing the performance on dev10d of CRF systems for direct sub-word-level confidence estimation. Curves for the baseline system, single-feature CRFs (LAPR and LSAPR) and full-featured CRFs are shown.*

false alarm probabilities of 40% the CRF-based systems yields consistently improved performance. The single-feature CRF which makes use of the LSAPR feature outperforms both the baseline, and the equivalent CRF system using LAPR across the full range of operating points. This is true for both the dev10d and eval09ns datasets. The NCE-based analysis however showed that the single-feature CRF using LSAPR performed worse on the dev10d dataset. The DET performance however suggests that the LSAPR predictor features are indeed generally better indicators of confidence for sub-words. Although the LSAPR feature uses posterior scores based on the same word-level acoustic and language model scores as LAPR, the fact that these scores are re-normalised over an appropriate set of sub-word arcs using the sub-word timing means that these scores are well-defined on the sub-word level. This is a pleasing result, showing that the use of sub-word-level predictor features can yield improved

performance over a system using the word-level LAPR predictor feature. The full-featured CRF system shows improvements in performance over all other systems, across all operating ranges for the eval09ns dataset. This is true for the majority of operating points on dev10d, except at very high false alarm probabilities, where performance is marginally worse than that of the LSAPR-based CRF. At a false alarm probability of 80%, the full-featured system shows an absolute decrease in miss probability of roughly 1%, and reductions of 5% to 7% at a false alarm probability of 1%.

8.4.2 *Hidden-state CRF*

The use of hidden states in a CRF model makes it possible for additional "hidden" sequence structure to be modelled. In this work, the values a hidden state variable can take are constrained to increase from left to right and return to 0 when there is a change from one output label to the next. On the sub-word level, it is believed that the structure that may be captured will be that of longer-span regions of consecutive sub-word units which are correct or incorrect. This is related to that of the word-level confidence regions discussed when utilising hidden states in the word-level model (see Section 7.4.6). One difference is that these regions are naturally expected to be longer, consisting of longer spans of sub-word-level observations.

The results obtained when applying hidden-state CRF models to the direct sub-word level confidence estimation task are shown in Table 8.2 on the facing page. Outcomes of statistical significance tests carried out for particular systems which represent interesting contrasts are also shown.

As can be seen in Table 8.2 on the next page, including hidden states in the confidence estimation model yields considerable improvements in NCE performance. Considering a configuration in which the word posterior (LAPR) and two sub-word posteriors (ASWP and LSAPR) are used by the models (system 2), the relative improvements over a standard linear-chain CRF are 26% and 10% relative for dev10d and eval09ns respectively. These improvements overshadow those that are observed when including additional predictor features in the model (see Table 8.1 on page 142). The results of significance testing carried out for this system against the baseline (which it outperforms by 0.089 absolute in NSWCE), show that the results are indeed significant, with a p-value below 0.001. It is also seen that the hidden-state system outperforms the baseline on almost all (95%) of the snippets in the audio. An additional significance test was carried out to investigate the significance of the performance improve-

System		dev10d		eval09ns	
Num	Description	NSWCE	UMAD	NSWCE	UMAD
	SWDT Baseline (LAPR)	0.208	13.96	0.266	6.88
①	SW CRF: LAPR+ASWP+LSAPR	0.263	11.68	0.310	6.21
②	SW HCRF: LAPR+ASWP+LSAPR	0.309	11.83	0.343	6.29

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
System ①	Baseline	0.049	0.001	85.8
System ②	Baseline	0.089	0.001	95.2
System ②	System ①	0.040	0.001	92.7

Table 8.2 *Results and evaluation of hidden-state CRF (HCRF) systems which operate solely on the grapheme level, to estimate grapheme confidence scores. Models are evaluated in terms of NSWCE and UMAD on dev10d and eval09ns. All continuous predictor features are represented using a spline approximation, over 8 evenly-spaced intervals, unless indicated otherwise. Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNSWCE = average difference between NSWCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

ments achieved when comparing a standard linear-chain CRF (system 1) to a comparable hidden-state CRF (system 2). This contrast shows that the improvement of 0.04 in NCE achieved by the hidden-state system is also statistically significant, and these improvements are still observed over as much as 92.7% of the snippets in the audio.

An analysis of the errors that result when applying a decision threshold (at 0.5) to the combined output of both datasets for the systems shown in Table 8.2 is also carried out. This analysis shows the hidden-state CRF yields a relative reduction in error rate of 8.2% over the baseline system. This reduction is also 1.6% larger than that for the full-featured linear-chain CRF. These reductions in error add further weight to the improvements observed in terms of NSWCE scores. The results obtained for the hidden-state system are encouraging, yielding some of the largest improvements reported in this work. This serves to support the use of hidden-state CRF models for confidence estimation in general, as this approach was also shown to yield improvements of a similar nature for the word-level task.

8.4.2.1 *Analysis of Detection Error Trade-off Curves*

The performance of the hidden-state CRF model for direct sub-word-level confidence estimation is evaluated using DET curves. The plots in Figures 8.4 and 8.5 on the next page show the performance (on eval09ns and dev10d respectively), of this hidden-state approach contrasted against the baseline and a linear-chain CRF. These DET plots are analysed in the discussion which follows.

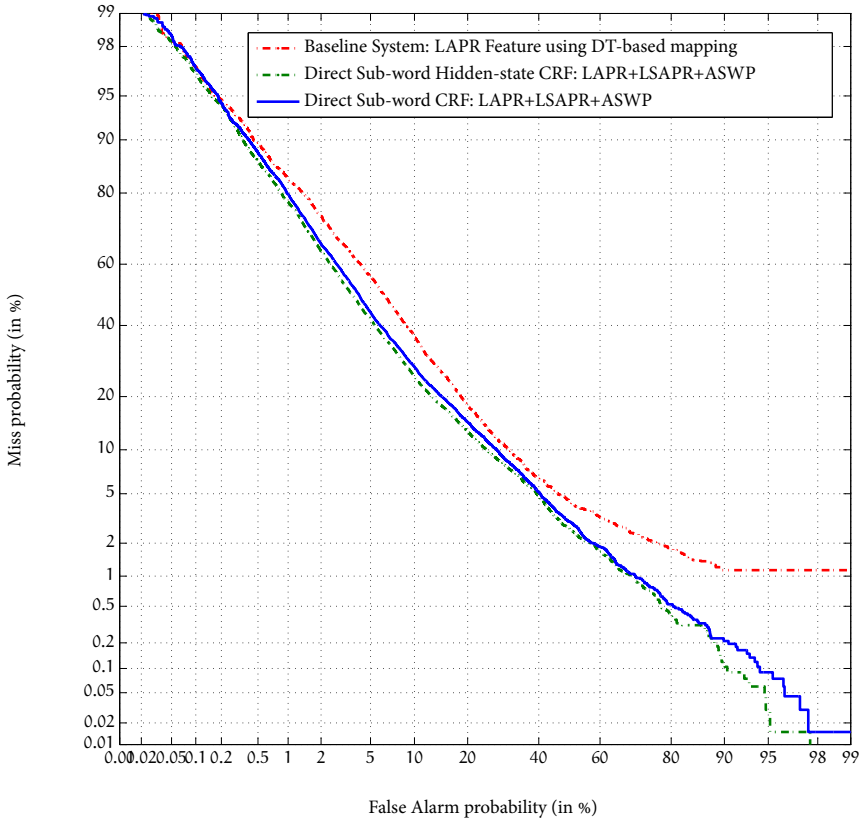


Figure 8.4 *DET curves showing the performance on eval09ns of the hidden-state CRFs for direct sub-word-level confidence estimation. Curves for the baseline system and a CRF with no hidden states are included in the plot for comparison.*

The DET curves plotted in figures 8.4 and 8.5 on the next page clearly show that the hidden-state CRF system outperforms an equivalent system with no hidden states and the baseline system across all operating points. This is particularly true on the dev10d dataset, where the margin of this improvement is considerably larger across all operating points. This may be an effect related to the fact that this dataset contains more errors, such that there is more of a distinction between the regions of confidence

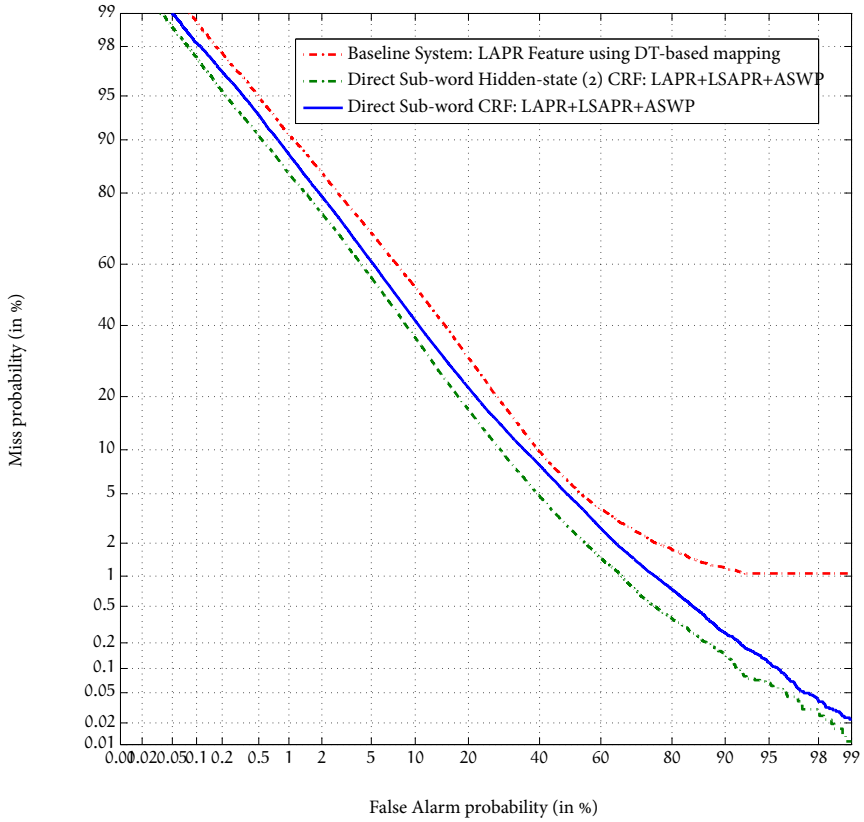


Figure 8.5 *DET curves showing the performance on dev10d of the hidden-state CRFs for direct sub-word-level confidence estimation. Curves for the baseline system and a CRF with no hidden states are included in the plot for comparison.*

in the hypothesised word being correct, and regions where these are indeed incorrect. This analysis is also supported by the NCE scores achieved by these systems, as the relative improvement on dev10d is 16% larger on dev10d than it is on eval09ns. The improvements observed on dev10d in this DET curve plot, signified by the large margin separating the linear-chain and hidden-state systems, are the largest observed in all experiments carried out for confidence estimation in this work.

8.4.3 Word-targeted Sub-word Confidence Estimation

In Section 8.4.1, sub-word-level information was used to estimate sub-word-level confidences directly. In the word-targeted approach, the CRF model also operates on the sub-word-level. However, the final output is desired to be defined on the word level. The premise in this approach is that sub-word-

level information can be exploited to improve word-level scores, and the best method through which this information can be utilised is to perform modelling at the sub-word level itself. This is different from an approach where, for instance, sub-word information is averaged up to generate word-level representations of this data in the input. In the proposed approach, the conversion to a word-level representation is instead carried out at the output stage.

As was mentioned in the motivation for the word boundary features (Section 8.3.1), in order for the sub-word CRF model to produce sensible word-level scores, it must be able to distinguish between successive words. If the model is not supplied with information pertaining to the segmentation of the sub-word-level observations into their corresponding word-level representation, it is not capable of capturing crucial characteristics of the multi-level (sub-word/word) problem. As an illustration of this issue, a situation is considered in which the model predicts that a sub-word observation is likely to be assigned the label `Incorrect`. This implies that the entire word which comprises this single sub-word and other sub-words, should also be labelled as being incorrect. However, without having information on where the word boundaries are, there is no way for the model to capture the fact that some of the preceding and following sub-words belonging to the same word should effectively be assigned the same label. Furthermore, the sub-word immediately following this incorrect sub-word (while still in the same word) might have a high probability of being correct. Assigning the `Correct` label to this sub-word should however not be allowed, as there is already a sub-word which is likely to be incorrect within the current word, which would result in the entire word therefore being incorrect. This is effectively a problem of missing context. The effect this has is that individual sub-word confidences may over-estimate the true confidence in the overall word.

In compensating for these effects using the word boundary feature functions, an additional input feature to the confidence estimation model is required. This input is a variable that indicates whether a particular observation is at the start of a word or within a word. The output label set is also extended to encode this word boundary information, such that the transitions between and within words can be captured. These modifications result in a model which has adequate knowledge of the locations of word boundaries in the sub-word-level sequence.

During test, the marginal probabilities for all labels which are derived from the base word-level `Correct` label are considered as contributing to the positive evidence for a sub-word being correct

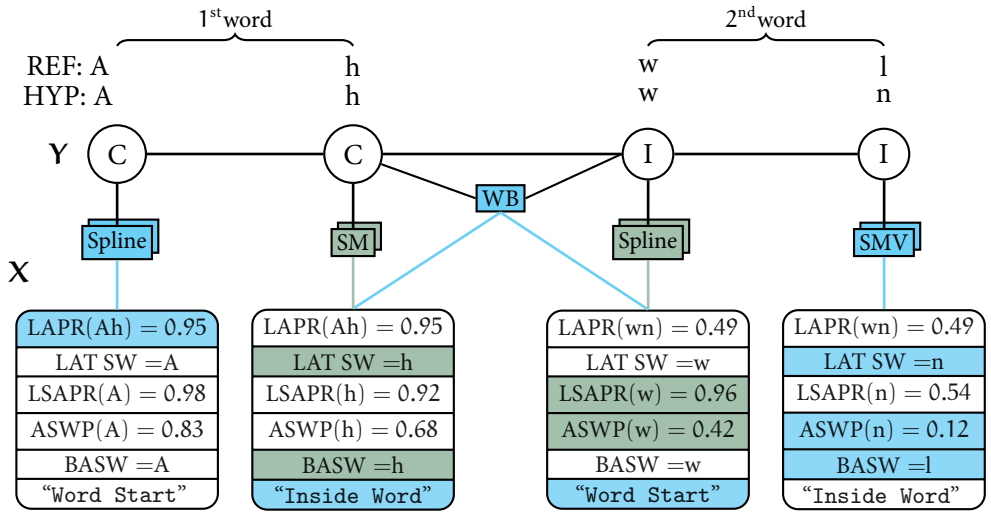


Figure 8.6 *Figure illustrating word-targeted sub-word-level confidence estimation using CRF models, with feature functions shown acting on appropriate dimensions of the feature vector.*

(i.e. word-boundary and within-word versions of the label `Correct`). These marginal probabilities are summed up for each sub-word. This yields a sequence of sub-word-level confidence scores for a given utterance. Finally, these scores are averaged over the constituent sub-words of a word to yield the desired sequence of word-level confidence scores. The accuracy of these confidence scores is evaluated in the same way as that employed in Section 7.4, using the word-level normalised cross entropy (NCE) and utterance-level mean absolute deviation (UMAD). This makes it possible for direct comparisons to be made with word-level confidence estimation systems.

An illustration of the approach taken for word-targeted sub-word-level confidence estimation is shown in Figure 8.6. There is one primary distinction between this figure and that illustrating the direct sub-word task in Figure 8.1 on page 139. Here, both the graphemes in the incorrect word are to be assigned the label `Incorrect`, regardless of whether the hypothesised grapheme itself is indeed correct. This is due to the fact that for the word-targeted task, the desired classification of sub-word observations into `Correct` and `Incorrect` observations is defined on the word-level. The word boundary `WB` feature functions are also shown in this figure, as they are important in the word-targeted approach. These feature functions act based on the value of an additional predictor feature in the input feature vector $\mathbf{X}$ which indicates whether the sub-word is at a word boundary. The additional indicator features are shown in the figure, and take the values "`Inside Word`" and "`Word Start`". With this inform-

ation, the WB feature functions are used to capture the word-level transition structure, by modelling transitions between output labels which may occur on word boundaries. The remaining `Spline`, string match (`SM`) and string match with value (`SMV`) feature functions are utilised in the same way as in the direct sub-word-level confidence estimation task.

In order to investigate the effect of modelling at the sub-word level, word-level representations of the sub-word predictor features and equivalent feature functions are utilised within a word-level confidence estimation system for comparison. The sub-word-level ASWP predictor feature is represented at the word level by averaging these scores over each word. This averaged predictor feature is referred to as AASWP (see equation 7.8). The SM feature function is used to match the word-level sequence of graphemes (comprising a word), assigned by the underlying system and the MLP recogniser. The SMV feature functions are used to match the aforementioned grapheme sequences, and return the value of the AASWP predictor feature.

8.4.3.1 Results

The results of experiments using CRF models which operate at the sub-word level, in which both word and sub-word level information are combined to ultimately estimate word level scores (word-targeted sub-word CE), are presented in Table 8.3 on the facing page.

The first interesting contrast in Table 8.3 on the next page is in comparing the word-level CRF with sub-word CRF systems with and without WB feature functions. These are the first lines within each sub-group shown in the table (systems 1, 5 and 9). Each of these systems makes use of the LAPR predictor feature in isolation. In comparing these systems it is clear that the word boundary feature functions (WB) are indeed important, as the sub-word system without these feature functions (5) is not able to approach word-level performance. This is to be expected, as systems without these feature functions have no explicit knowledge of word boundaries. The fact that there is a mismatch between the level at which the model operates, and the level at which the output is desired can therefore not be compensated for by the model which does not include these feature functions. As the evaluation is carried out at the word level, such a model is therefore not matched to the task. Significance testing of the results for this system (5) show that these results are significant, with the p-value lower than 0.001. As the system operates on the sub-word-level, it is inherently quite different from the baseline

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	Baseline	0.325	12.64	0.356	6.98
①	WCRF: LAPR	0.339	11.33	0.359	6.64
②	+SM	0.342	11.25	0.359	6.64
③	+SMV (AASWP)	0.346	11.09	0.360	6.60
④	+AASWP	0.346	11.06	0.361	6.58
⑤	SWCRF: LAPR	0.305	10.93	0.318	6.59
⑥	+LSAPR	0.313	10.75	0.322	6.54
⑦	+ASWP	0.318	10.56	0.324	6.55
⑧	+LSAPR+ASWP	0.319	10.61	0.326	6.54
⑨	SWCRF: WB + LAPR	0.338	11.48	0.362	6.58
⑩	+ LSAPR	0.344	11.19	0.364	6.54
⑪	+ SM	0.348	11.04	0.361	6.60
⑫	+ SMV(ASWP)	0.352	10.92	0.364	6.57
⑬	+ ASWP	0.353	10.93	0.367	6.53
⑭	+ LSAPR + ASWP	0.354	10.93	0.368	6.52

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
⑤	Baseline	-0.031	0.001	30.3
⑨	Baseline	0.008	0.001	59.7

Table 8.3 *Word-targeted sub-word-level confidence estimation experiments. Models are evaluated in terms of NCE and UMAD on dev10d and eval09ns. All continuous predictor features are represented using a spline approximation, over 8 evenly-spaced intervals. Numbers are provided for each system to aid in the discussion of the results. Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNCE = average difference between NCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

system, so this result is not surprising. The substantive analysis in terms of degraded NCE performance (−0.031), and a low proportion (30.3%) of snippets on which this system outperforms the baseline, show that this statistically significant difference is such that the CRF system is worse (at least as far as the NCE-based evaluation). A sub-word-level system which includes the WB feature functions and the LAPR predictor feature (system 9) performs as well as an equivalent word-level CRF system, but

no better. However, it is interesting to note that there is a decrease in UMAD performance on dev10d when including the word boundary information through the WB feature functions. This suggests that when the alignment of scores to words is not a key component of the evaluation, the original sub-word model scores are actually quite accurate, and are thus capable of yielding decent performance when averaged over at the utterance-level. The improvement in terms of NCE performance (0.0008) for system 9, whilst small, is shown to be statistically significant (p-value less than 0.001). This system is also able to outperform the baseline on over half of the snippets (59.7%).

Including the grapheme-level predictor feature (LSAPR), which is truly defined on the sub-word level, results in an improved system (10). This system shows improvements in both the NCE and UMAD performance over the single feature (LAPR) system (9). The inclusion of the SM feature functions to match sub-word unit identities yields slightly larger improvements (system 11), suggesting that the information from the alternative recogniser is indeed useful, and can contribute in the confidence estimation process. This gain is larger than that observed in the word-based CRF equivalent (system 2), where grapheme sequences are matched at the word-level, rather than individual graphemes. This result highlights a benefit of modelling at the sub-word level, as the model is able to make better use of the information pertaining to competing grapheme hypotheses at this level, rather than grapheme sequences at the word level. Through the inclusion of the ASWP predictor feature as the continuous value returned by the SMV feature functions (system 12), further gains are achieved. Once again, the word-based system (2) shows improved performance over one in which this information is not utilised, but does however not perform as well as the sub-word-level model. Representing the ASWP predictor feature with spline feature functions instead (system 13) yields improved performance over the system making use of the SMV feature functions to represent the first order moment of this predictor feature. Finally, the inclusion of the LSAPR feature (system 14) yields slight performance gains on both data-sets. As this predictor feature is very closely related to the word-level posterior (LAPR) feature in this task, it does not contribute a great deal when included in the model. Nevertheless, this results in the best system of this type, which clearly outperforms the equivalent word-level system (4). This system shows relative improvements over the baseline on dev10d of 8.9% and 13.5% in NCE and UMAD respectively.

Despite the fact that the LAPR predictor feature is essentially a word-level measure, this predictor

feature may be incorporated into the sub-word-level system. Results shown in Table 8.3 on page 155 prove that doing so is indeed useful. This makes sense, given the task is actually still that of assigning word-level confidence scores. It is therefore worth investigating the effect of including additional word-level predictor features in the model, such as those utilised in the word-level confidence estimation systems detailed in Chapter 7. Naturally, the values for these predictor features will not vary over the graphemes corresponding to the same word, and are therefore repeated in each grapheme-level set of predictor features for a given word. Results of experiments in which additional predictor features, the majority of which are defined on the word level, are incorporated into the sub-word-level framework are shown in Table 8.4 on the next page. The sub-word-level model is contrasted against a comparable word-level model, as well as the word-level decision tree baseline. Additional predictor features are incorporated into the model with first-order moment features (M1) for these experiments. This reduces the number of parameters within the model, thus facilitating rapid training of the CRF models. The results of significance tests performed on particular systems of interest are also provided.

The results presented in Table 8.4 on the following page clearly show that word-level predictor features improve confidence estimation performance of the sub-word models. The base model used for these contrasts is the sub-word-level model which makes use of spline feature functions applied to each of the LAPR and ASWP predictor features (system 1). A set of three predictor features (WCR, LAS and ACPOST), which are collectively related to the acoustics and stability of the hypotheses, is added to the model (system 2). This yields relative improvements in NCE of 1% and 2% on dev10d and eval09ns respectively. There is also a slight improvement in the UMAD metric for eval09ns. The language model posterior LALMP proved useful in the word-level system, and is therefore included in the model (system 3). This system yields improvements on dev10d, but not on eval09ns. This may be attributed to the fact that with fewer errors in the eval09ns data, there are fewer instances where there is likely to be contention between the acoustic and language models, as the recogniser is more certain about its hypotheses. The degree to which the language model is confident in a hypothesis is the information encoded by this predictor feature, which is naturally less useful when there is less contention. The inclusion of some longer-range contextual information above that of the word-level, through the TF*IDF predictor features, is also shown to yield some gains (system 4). It is interesting that even with a model defined at a level of granularity significantly lower than that at which the TF*IDF

System		dev10d		eval09ns	
Num	Description	NCE	UMAD	NCE	UMAD
	Baseline	0.325	12.64	0.356	6.98
①	SWCRF: WB + Spline 8E (LAPR)	0.338	11.48	0.362	6.58
②	ABOVE+Spline 8E (ASWP)	0.353	10.93	0.367	6.53
③	ABOVE+WCR+LAS+ACPOST	0.359	10.95	0.374	6.48
④	ABOVE+LALMP	0.363	10.83	0.374	6.48
⑤	ABOVE+TFIDF	0.369	10.71	0.376	6.60
⑥	ABOVE+HA+LPS(3)+LPS(2)+WS(1)	0.371	10.68	0.376	6.64
⑦	WCRF: BEST10FEAT+AASWP+ABASWP	0.367	10.75	0.376	6.46

Systems		dev10d+eval09ns		
A	B	ΔNCE	p(<)	SIP(%)
⑥	Baseline	0.032	0.001	75.8

Table 8.4 *Word-targeted sub-word-level confidence estimation experiments, in which additional word-level predictor features are utilised. Models are evaluated in terms of NCE and UMAD on dev10d and eval09ns. For the sub-word systems, the LAPR and ASWP predictor features are represented with spline feature functions using 8 evenly-spaced knots, with the remaining features represented using first order moment feature functions. Results of significance tests on selected systems are also shown, evaluated on the combined dev10d and eval09ns datasets. ΔNCE = average difference between NCE scores of systems, p(<) = p-value for test, SIP = snippet improvement percentage.*

is computed (i.e. graphemes vs. segments of audio), this information is still utilised effectively by the model. Including the entropy in the scores generated by the alternative recogniser, as well as the deviation metrics result in a small improvement in NCE on dev10d (system 5). These predictor features were seen to yield larger gains in the word-level system. The reduced impact of these features may be a consequence of the information in these scores being diluted when repeated over graphemes to obtain their approximate forms on this level. The final sub-word level system shown in this table (system 6), makes use of the full set of predictor features, including the word-level lattice posterior stability (LPS) and word stability (WS) predictors. This yields another incremental improvement in performance. The results of this system are shown to be statistically significant, with a p-value of less than 0.001. It is also able to outperform the baseline in 75.8% of the snippets in the audio, which is 16.1% absolute more than in the system without the additional word-level predictor features. This final sub-word-level

system makes use of the same information as the word-level system shown in the table for comparison (system 7). In the word-level system however, spline feature functions are applied to the predictor features. In comparing the sub-word-level system to the word-level equivalent, this system yields a 1% relative improvement on the dev10d dataset, with no gains seen on eval09ns. The use of spline approximations for the additional predictor features may however yield improvements over the word-level system.

8.4.3.2 Analysis of Detection Error Trade-off (DET) Curves

The performance of the word-targeted sub-word confidence estimation systems is evaluated in terms of their respective DET curves. A selection of these are shown for the eval09ns and dev10d datasets in Figures 8.7 on the next page and 8.8 on page 161, and are analysed in the discussion which follows.

The baseline considered in the DET plots of Figures 8.7 on the next page and 8.8 on page 161 is the same as in the standard word-level system. This baseline is the decision-tree based segmentation of the interval for the LAPR predictor feature, with a subsequent piecewise-linear mapping being applied. For both datasets, the CRF systems which make use of the LAPR predictor features in isolation show clear improvements in mid to high-range false alarm probabilities over the baseline system, and only yield performance inferior to that of the baseline at very low false alarm probabilities. This is true of both the system which includes the word boundary (WB) feature functions, and that which does not. In fact, the performance of these LAPR-only systems is very similar, with there being some improvements in the system including WB on dev10d. This is an interesting observation, as the NCE-based analysis suggested there was a much larger performance difference between these systems (almost 10% relative on dev10d). An explanation for this is that the DET-based analysis evaluates the systems at different confidence thresholds and is effectively concerned with the number of errors which result at each such threshold. The NCE score is however influenced to a large degree by the actual value of the confidence scores. The further the scores deviate from the ideal values, the more they incur penalties, irrespective of whether or not these scores may or may not actually have resulted in an error at a given threshold. While the system without word boundary feature functions might produce good estimates of confidence, these scores may not change sharply on word boundaries. It is expected that this has an adverse effect on the NCE-based evaluation, with it being sensitive to the deviation of scores from their ideal

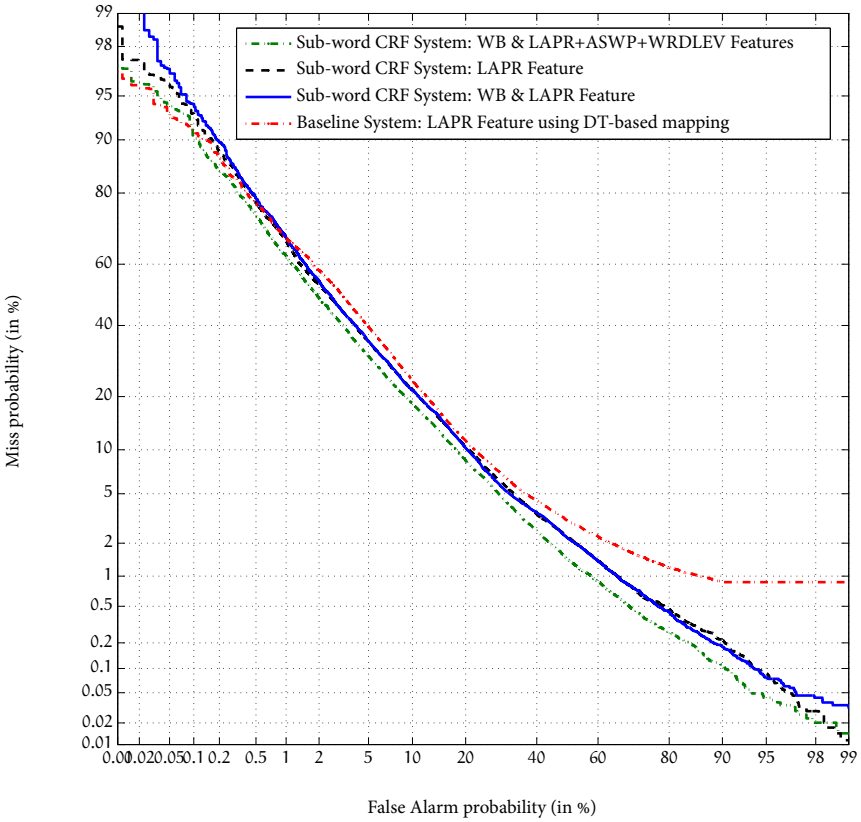


Figure 8.7 *DET curves showing the performance on eval09ns of CRF systems for word-targeted sub-word-level confidence estimation, in which multiple predictor features are combined. Curves are shown for the baseline system, LAPR-based CRFs with and without word boundary feature functions, and the full-featured CRF system.*

(1/0) values. This analysis implies that if confidence scores are to be used purely in a decision setting (i.e. to accept or reject a word hypotheses as correct), the influence of word-level boundary information is not as important. However, if the scores themselves are to be compared with scores from other systems, or used in decision making further downstream, it is useful to include this information so as to improve overall score accuracy.

Combining additional sub-word and word-level predictor features to yield the full-featured model results in further gains in performance across all operating points, as was observed on the word-level. The inclusion of the additional predictor features also serves to improve performance for the range of operating points at which the false alarm probability is low. Thus improving upon the system that made use of the LAPR predictor feature in isolation, which yielded poor performance this region. This effect

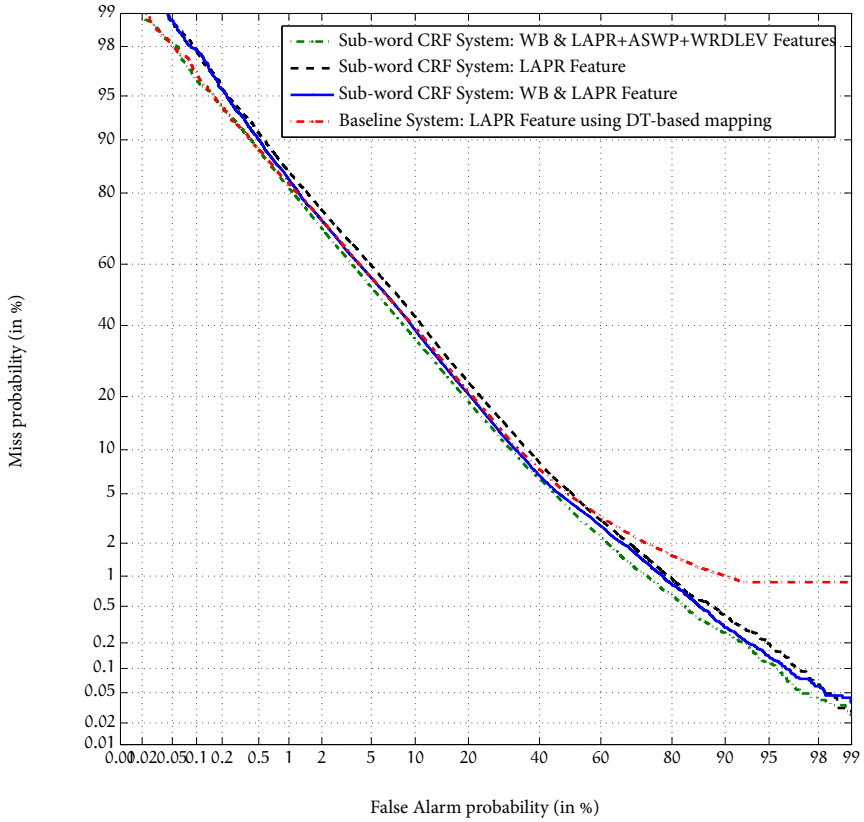


Figure 8.8 *DET curves showing the performance on dev10d of CRF systems for word-targeted sub-word-level confidence estimation, in which multiple predictor features are combined. Curves are shown for the baseline system, LAPR-based CRFs with and without word boundary feature functions, and the full-featured CRF system.*

was also observed in the word-level system evaluation (for an explanation see Section 7.4.4.1). The DET plot for a word-level system equivalent to the full-featured sub-word system is largely indistinguishable from that of the sub-word system, indicating that the performance achieved in the two approaches is largely equivalent. The DET curve for this word-level system is therefore omitted from the DET plots for the purpose of clarity.

8.5 Summary Discussion

In this chapter, confidence estimation models which make use of information defined on the sub-word (grapheme) level are developed. The architecture and framework applied to the task of word-level con-

fidence estimation is refined and extended as required for this domain. Two tasks are considered, the first being that of directly estimating confidence scores for sub-word units (direct sub-word confidence estimation). The second task is that of improving word-level scores using sub-word information (word-targeted sub-word confidence estimation). The CRF-based approach to sub-word-level confidence estimation of both types is a novel contribution of this work. In both these tasks, the proposed approaches are shown to outperform sensible baseline systems considerably, by combining word and sub-word information effectively.

8.5.1 Direct Sub-word Confidence Estimation

For direct sub-word confidence systems, the task is somewhat different from the word-targeted approach. There is no mismatch between the form of the output of the CRF and the output required for the task. As the model therefore acts on the desired level inherently, characteristics of the sub-word-level problem impact the confidence estimation model directly to improve scores. This direct relationship ensures aspects of the problem can be captured by the model without being obfuscated through the mapping to the word level.

As in the case of word-level confidence estimation, the CRF model (with automatically placed spline points applied to the word-level posterior (LAPR)) is shown to yield improvements over the decision tree baseline for sub-word level confidence. The approach taken in this work is thus immediately shown to be useful. Combining the lattice-based sub-word posterior (LSAPR) with the word-level LAPR posterior in these models resulted in one of the most substantial improvements in performance seen for feature combinations in this work. Relative gains in NSWCE over the LAPR baseline for this two-way combination are 20.7% and 14.7% for dev10d and eval09ns respectively. Relative improvements over a single-feature CRF system using LAPR of 16.7% and 19.1% for dev10d and eval09ns are also achieved. The reason these features are complimentary may be explained by considering that if a given grapheme is within an incorrect word (which might result in a lower LAPR score), the true likelihood of any grapheme within that word also being incorrect should naturally also be higher. This is due to the fact that at least one sub-word (potentially the current one), within the word must be incorrect for the entire word to have been incorrect. The information encoded in the LAPR predictor feature can therefore effectively be weighed off by the model against the evidence encoded in the sub-word

posterior score (LSAPR).

The sub-word level scores produced by an alternative recogniser for the grapheme hypothesised by the underlying system proved to be useful. Whilst including the raw score for the best alternative grapheme yielded no improvements over a combined system of the other features, learning how this score is distributed for each of the graphemes yielded some gains. This is slightly counter-intuitive, as these scores are not generally related to the sub-word hypothesis. There is however expected to be some overlap between the two different hypotheses, such that when this is the case, this information will contribute to the confidence estimation process.

Making use of hidden-state CRFs on the sub-word level lead to sizeable gains, of a larger degree than those observed in the word-level model. In fact, the best sub-word-level confidence estimation model developed in this work is one which makes use of hidden states. The hidden-state system achieved large relative improvements in NCE over the decision tree baseline of 41.6% and 28.9% for dev10d and eval09ns respectively. As was the case for word-level confidence estimation, these improvements are attributed to the ability of hidden-state models to capture information pertaining to regions of confidence. In this case these regions correspond to sequences of correct or incorrect sub-words. As no word-level constraints are applied, this allows the model to learn this structure irrespective of the word boundaries. These regions may begin or end at any point within a word. This is important given the nature of the sub-word-level task, and ultimately enhances the predictive power of the confidence estimation system.

The improvements in performance achieved through the novel application of sub-word-level CRF models for the direct sub-word confidence estimation task proposed in this work are some of the largest improvements reported in this thesis. These improvements are further amplified when hidden-state CRFs are applied to capture additional characteristics of the sequential nature of errors. This serves to support one of the themes of this thesis, which is that the sequential nature of the confidence estimation problem can be exploited to improve the accuracy of confidence scores.

8.5.2 Word-targeted Sub-word Confidence Estimation

It is shown that for word-targeted systems, the mismatch in the level at which the model inherently operates and that of the task must be addressed to yield sensible word-level output. This problem is dealt

with successfully as part of the model itself, by making use of the flexibility of the CRF-based framework to engineer feature functions which enforce word-level structure in the sub-word model. With the models enhanced in this way, they are capable of exploiting sub-word information in combination with word-level information to yield improved confidence scores.

The sub-word-level predictor features extracted from a completely separate grapheme-based recogniser proved to be useful. This is in line with the understanding that information from "different" sources is typically useful in combination with predictor features extracted from the underlying system. The fact that modest gains resulted when combining the lattice-based sub-word posterior (LSAPR) with the overall word posterior (LAPR) further supports this understanding. This may be attributed to the fact that for the word-level task, no additional useful information is captured by LSAPR not already present in LAPR, and these predictor features are therefore not different in a way which contributes new information to the system.

Feature functions were developed that match the underlying and alternative sub-word unit hypotheses. While modelling this information through the use of these feature functions yields incremental improvements, their utility is overshadowed by the gains obtained when more informative predictor features are included in the model.

The primary result of interest for these systems is that in order to leverage information extracted from a sub-word source (and thereby yield improved confidence scores), modelling should in fact be carried out at this level. Although this implies additional modelling effort, the results show that converting such information to word-level representations in order to make use of this in a model operating at a higher-level is not as effective.

Chapter 9

Confidence in Keyterms

The framework for confidence estimation described in this thesis was initially developed for the task of annotating a full transcription output by an ASR system (e.g. the 1-best hypothesis) with confidence scores. Further developments made it possible to annotate arbitrary hypotheses with confidence scores, where these hypotheses were typically sourced from the output of different stages of an ASR pipeline. In this chapter, the definition of arbitrary hypotheses is extended further to include that of hypotheses for keyterms[1] in a spoken term detection (STD) system. The problem of assigning detection scores to keyterm hypotheses is consequently reformulated as a confidence estimation task, which plays to the strengths of the framework proposed in this work. One of these strengths is the ability to combine multiple information sources in estimating confidence scores. Another benefit of using this framework is that it supports the development of arbitrary feature functions, such that the predictor features may be utilised to best effect by the model. Both these aspects are exploited in taking a CRF-based approach to what will be referred to as *keyterm confidence estimation*, in a novel contribution of this work.

The scores assigned to detection hypotheses are a crucial aspect of performance in a keyterm detection system. The accuracy of these scores has a large impact on performance, as a decision is typically made on whether to reject or accept keyterm hypotheses based on a thresholding decision applied to these scores. The performance of a keyterm detection system in which the detection scores are estimated using the CRF-based framework is therefore evaluated. This allows for further development and

[1] Keyterms are terms to be detected that consist of one or more words.

understanding of the framework developed in this work. It also serves as a concrete example of the applicability of the overall approach to confidence estimation.

A *hybrid* keyterm spotting system, consisting of word and sub-word-level components, is developed as part of this work. This chapter will proceed by detailing each component of the hybrid keyword spotting system, where the predictor features relevant for each of these components are also presented. Thereafter, extensions to the CRF-based infrastructure developed specifically to address aspects of the keyterm detection task are detailed. Finally, the results of experiments in applying the keyterm confidence system are presented and discussed.

The contributions presented in this chapter are based on published work detailing a CRF-based approach for estimating keyterm confidence and compensating for score normalisation (Seigel *et al.* 2013).

Related work in Ou and Luo (2012) considers the estimation of keyterm detection scores with CRF-like models. There are a number of differences between that work and the CRF-based approach taken here. The first major difference is that the CRF models of the aforementioned work are not truly sequential at the state level, and are in fact more closely related to maximum entropy models, as there are no transition feature functions defined for the label states. The use of `CRFTK` also made it possible for a number of useful continuous lattice based predictor features to be incorporated into the models effectively using spline feature functions in this work. In the related work, use could only be made of discrete representations for a limited number of features available from the confusion networks from which keyterm hypotheses are extracted. The direct model-based approach proposed for accounting for term dependence also sets the CRF-based approach of this work apart from the related work.

9.1 The Hybrid Keyterm Spotting System

In the approach taken to carry out spoken term detection (STD) in this work, the audio data is decoded once by the ASR system to generate lattices. These lattices represent a static search space, within which occurrences of keyterms are to be discovered. An index may be built once (offline), which consists of all the words in the vocabulary and the times that they are hypothesised to have occurred in the audio data. This is commonly referred to as a pre-indexed approach. Techniques in which the ASR decoding

phase uses knowledge of the keyterm list to adapt the search Zhang *et al.* (2012) may yield improved performance over the pre-indexed approach. However, changes to the keyterm list require that the audio be re-decoded by the ASR system in such an approach. In this work, a pre-computed index is not built once for all possible words in the vocabulary. Instead, a search is carried out over the current keyterm list in the recognition lattices output by the ASR system. The primary reason for this being that along with every hypothesised keyterm occurrence, a set of predictor features is extracted from the lattice. These features are required by the confidence estimation model. Performing a combined search and extraction step for a given keyterm list is therefore a sensible approach in the context of this work. In addition, this approach has the benefit that should the keyterm list change, the audio need not be re-decoded. This approach is therefore more scalable.

The hypothesis space within which keyterms are to be detected is restricted by the size of the ASR system lattices and the vocabulary used by the system. This problem is addressed by generating lattices and carrying out keyterm search at both the word and sub-word levels, as was investigated in James (1996); Logan *et al.* (2002); Yu *et al.* (2005). In spoken term detection systems, the trade-off between the number of false positives and false negatives generated is an important factor in system design and implementation. The word and sub-word systems represent different operating points within this trade-off, with the combination of the two yielding a hybrid solution with the benefits of the individual systems.

9.1.1 Word-Level Keyterm Spotting

Keyterms to be detected may consist of more than one word. Therefore, in performing keyterm search in the word-level system, all occurrences of words which constitute keyterms are considered as being partial keyterm hypotheses. Each such partial hypothesis is associated with a start time and a set of relevant predictor features. Given the set of partial hypotheses, individual word hypotheses are grouped into keyterm hypotheses in cases where certain constraints are met. These constraints enforce the correct ordering of the words within the keyterm, as well as a maximum allowable time interval between two consecutive words. This results in a set of hypotheses for complete keyterms within audio recognised by the ASR system.

The word-level approach yields a fairly low number of keyterm hypotheses. This is largely due to

the fact that the ASR output lattices represent a constrained hypothesis space, from which keyterms may have been pruned away during search. Furthermore, as the search performs exact matching, only those keyterms for which all constituent words are covered by the ASR vocabulary may be discovered. These factors result in a small number of accurate keyterm hypotheses. These hypotheses are useful, but result in the system only being able to reach a relatively short range of possible operating points.

As with CRF-based confidence estimation for 1-Best ASR hypotheses, a set of predictor features which is indicative of the quality of the hypothesis is required for each hypothesis generated by the underlying system. This is one of the motivating factors for developing a keyterm detection system in this work, as it is necessary to compute these features from the system lattices during the keyterm search phase. In the following sections, those predictor features relevant for the word-level system will be presented.

9.1.1.1 *Lattice-based Arc Posterior Ratio*

This feature proved to be effective for general confidence estimation, and is therefore used once again for keyterm confidence estimation. The only significant difference between the form expressed in equation 7.4, is in the word argument for which the posterior is to be computed. Here the LAPR score is not computed for a word within a one best sequence (W_i), but rather for a specific occurrence of a keyterm (or word comprising a keyterm), denoted as K. This predictor feature will henceforth be referred to as LAPR(K).

9.1.1.2 *Contextual Posterior Features*

Keyterm hypotheses are fundamentally different from those considered previously in the context of confidence estimation, where, for example, the 1-Best hypothesis is used. In the 1-Best case, this hypothesis is typically generated by carrying out Viterbi decoding. This implies that each word within this hypothesis is along a complete path of successive word hypotheses spanning the lattice, where this path results in the best overall likelihood. In keyterm detection, the presence of keyterms in the system output at all triggers a keyterm detection hypothesis. This detection trigger is not based on the score related with a detection or its location within a likely path through the lattice. Whilst the scores as-

signed to these hypotheses are crucial in the thresholding decision used in accepting or rejecting them, they are not used during the detection phase.

A means of capturing some idea of context for keyterm detections, particularly in terms of whether it lies on a likely path through the lattice is sought for use in this work. Predictor features are therefore proposed which represent the confidence of the ASR system in the words immediately preceding and following the keyterm/word itself (i.e. the context). For a lattice arc corresponding to a keyword hypothesis, the most likely preceding arc K' and following arc K'' within the lattice are found. These arcs represent the localised context for the keyterm or word within a keyterm. Applying equation 7.4 to these word arcs yields the contextual features $\text{LAPR}(K')$ and $\text{LAPR}(K'')$.

These features are particularly relevant in the word-level system. However, a similar set of contextual posteriors can be computed in the sub-word level system. In this case, the preceding context posterior corresponds to that of the parent arc of the first sub-word within the keyterm, and the following context posterior is that of the parent arc for the last sub-word of the keyterm.

9.1.1.3 Unigram Prior Features

Certain keyterms are more likely to occur in language than others. When audio is automatically transcribed using ASR systems, the fact that unlikely words are by definition assigned lower language model scores is desirable, and indeed useful. However, in keyterm detection, certain keyterms of interest may contain rarely-occurring named entities, or other words which are less common in language but very important to be able to detect. As a result of this, a predictor feature is sought which aims to provide some contrast to the LAPR predictor feature (which includes the language model scores). This predictor feature should effectively inform the model when confidence scores should be boosted or discounted to account for the aforementioned phenomenon. The feature proposed for this purpose is the prior probability of the keyterm. Unigram probabilities for words within a given keyterm are obtained from the language model of the ASR system and multiplied to yield the overall keyterm prior. The logarithm of this value is used as the unigram prior (UP) feature for a given keyterm. Naturally, this predictor feature is constant for all detections of a certain keyterm.

9.1.2 *Sub-word-level Keyterm Spotting*

The primary motivating factors for carrying out a keyterm search on the sub-word level are the following:

1. The nature of the search typically results in a large number of keyterm detection hypotheses. This is desirable for some applications in which the required operating point is one in which missing a keyterm incurs a higher cost than generating false alarms.

2. Terms which are out-of-vocabulary (OOV) for the recogniser used to decode the audio may be detected. This is particularly important for named entities, which tend to change over time. It is also relevant for words which are generally infrequent, but are of interest despite their low typical occurrence rate.

Due to the fact that these systems generate a large number of detection hypotheses, the scores associated with the detections are of critical importance. If a system were to generate a large number of detections to ensure a low miss rate, but be capable of predicting with high accuracy when these hypotheses are spurious, this operating point could be achieved with a low false alarm rate, which is ideal.

In order to perform keyterm search at the sub-word level, the word-level lattices are marked up with sub-word timing and acoustic score information. Using this information, keyterm search proceeds by initially considering all occurrences of sub-words in the lattice which constitute any of the keyterms as partial hypotheses. At this stage, predictor features are computed and attached to each such partial hypothesis. Thereafter, a clustering stage is carried out such that individual sub-word partial hypotheses are clustered together to form longer keyterm hypotheses. Constraints are applied during clustering to the partial hypotheses considered as candidates which form part of a complete keyterm hypothesis. These constraints specify that the partial hypotheses must occur in the correct order, and the time interval between them must be sufficiently short, such that they may effectively be considered as having come from the same word. In this process, sub-words which form part of different word-level hypotheses can be concatenated together as part of new keyterm hypotheses. This productive process results in a large number of new keyterm hypotheses, which would potentially not have been hypothesised in the word-level system.

9.1.2.1 *Lattice-based Sub-word Arc Acoustic Posterior Ratio*

The nature of the sub-word-based keyterm search implies that keyterms are pooled across many different *parent word arcs*. These parent arcs are the word-level arcs which the sub-word arc forms part of. The acoustic score for each sub-word arc is available, but there is no sensible way to represent grapheme-level language model scores. This fact may effectively be ignored, by using the language model scores for each parent arc as the language model score for each sub-word. This is the approach taken in Lo *et al.* (2004); Pan *et al.* (2007). However, a concern with this approach is how to normalise the word-level language model scores over the constituent sub-words effectively. In this work, a score similar to the LAPR, which is based solely on the acoustic scores for sub-words is therefore proposed. This feature, which shall be named the lattice sub-arc acoustic ratio (LSAAR) is computed from a sub-word lattice for a keyterm K by averaging over the N_K sub-words it contains as follows:

$$\text{LSAAR}(K) = \frac{1}{N_K} \sum_{i=1}^{N_K} \frac{\sum_{S \in \mathcal{I}} \delta(\text{sub-word}(S), K_i) p(S|\mathbf{O})}{\sum_{S \in \mathcal{I}} p(S|\mathbf{O})}$$

where K_i is the identity of the sub-word at index i of N_K sub-words in keyterm K, the sub-word arc S has the identity $\text{sub-word}(S)$, and $\mathcal{I}$ is the set of intersecting sub-word arcs over which the sub-word-unit acoustic model scores $p(S|\mathbf{O})$ are summed.

9.2 Direct Model-based Score Normalisation

A characteristic of most spoken term detection systems, is that the scores for keyterm detections are computed in a manner such that the resulting scores are not dependent on the actual keyterm to which they correspond. In decision making for keyterm detections (i.e. whether to accept or reject a given hypothesis), a global threshold is typically applied to these scores. This decision-making criterion asserts the assumption that a single threshold effectively implies the same level of confidence across all keyterms. Different keyterms may have vastly different characteristics, such as their length, frequency of occurrence, and the language model scores assigned to the keyterms. These factors result in the scores for different keyterms effectively falling into different ranges. The application of a single global threshold therefore corresponds to different effective confidence levels for different keyterms.

This issue may be addressed by modifying the evaluation process such that different thresholds are applied for each keyterm. Another set of approaches are aimed at mapping the scores in the system output before applying a single global threshold in evaluation. Rank-based and term-weighted normalisation (Miller *et al.* 2007; Vergyri *et al.* 2006; Zhang *et al.* 2012) have proven to be effective solutions of this type. Discriminative score mapping (Wang *et al.* 2012) is another approach which aims to achieve this normalisation through direct modelling.

One advantage of the CRF-based approach taken in this work, is in the flexibility of the models. Specifically, it is possible for arbitrary feature functions to be engineered and incorporated into the model framework with ease. A novel contribution of this work is that of directly addressing the score normalisation problem as part of the overall keyterm confidence estimation process with CRF models. This is achieved by engineering keyterm-specific feature functions.

The keyterm-specific feature functions developed in this work are used to effectively learn the distribution of a continuous predictor feature (such as the LAPR score) separately for each keyterm. These feature functions therefore take a literal feature as an argument (i.e. the keyterm identity in the data), and return the value of the desired continuous predictor feature, provided the literal value matches the template value for which the feature function is defined (i.e. a specific keyterm). The literal moment (LITM1) feature functions are defined as follows:

$$\text{LITM1}_{y,l}(\mathbf{Y}, \mathbf{X}[p]_i, \mathbf{X}[s]_i) = \mathbf{X}[p]_i \, \delta(\mathbf{X}[s]_i, l) \, \delta(Y_i, y)$$

where y and l are the template label and literal value (keyterm) for which the feature function is defined, $\mathbf{X}[s]_i$ and $\mathbf{X}[p]_i$ are the literal (a keyterm identity) and continuous feature values corresponding to the current observation and Y_i is the label for the current observation (i) in the label sequence $\mathbf{Y}$. The LITM1 feature functions therefore learn separate first-order moment statistics of a continuous feature, for each possible value the literal feature can take (i.e. the keyterm identities). It should however be noted that as these feature functions rely on the identity of the keyterms, changes to the keyterm list imply that a CRF model making use of these feature functions would have to be retrained for the new keyterm detection task.

9.3 Experimental Setup

Keyterm spotting experiments were carried out on a state-of-the-art recogniser which was built using data from the DARPA robust automatic transcription of speech (RATS) program for Arabic keyword (or keyterm) spotting (Graff *et al.* 2011). The data provided under this program consists of Levantine Arabic conversational telephone speech, which had been recorded and subsequently retransmitted over eight severely degraded communication channels. The recognition task is highly challenging. This is due to the high level of noise present in the audio, as well as the conversational nature of the language.

The recogniser described in Gales and Flego (2012) is used as the underlying system for keyterm detection in this work. For the recognition task, a limited amount of audio training data (between 2 and 2.5 hours) is available for each of the channels. Due to the limited amount of Leventine Arabic text available, language model training data was limited to the 1.6M tokens words available in the supplied transcriptions of the training data. Both bigram and trigram language models were built on this text, using Kneser-Ney smoothing (Ney *et al.* 1994) to compensate for the data sparsity problem. The audio front-end processing applied ultimately yields a 39-dimensional PLP-based feature vector for each frame. This feature vector is obtained by extracting 13 PLP cepstra (0^{th} order (energy) to 12^{th} order coefficients) from the audio, and additionally computing the first, second and third order derivatives (or deltas) for each. Thereafter, an HLDA projection (Kumar 1997; Liu *et al.* 2003) is subsequently applied to these features to reduce the dimensionality of the vector from 52 down to 39. The recogniser used is a word-based graphemic system. Cross-word triphone HMM acoustic models are trained using MPE, resulting in a total of 3k tied states. There are three consecutive decoding phases for this recogniser. In the first phase, link adaptive training (LAT) is applied. Here 128 CMLLR transforms are used to represent the channels and FE-CMLLR (Liao and Gales 2005) is used to compensate for the noise. In the second decoding phase (P2), unsupervised MLLR speaker adaptation takes place. The output of this phase is used as the supervision for speaker adaptive training (SAT) with CMLLR applied in the third phase (P3). This results in a link and speaker adaptively trained (LSAT) system.

It was found through experimentation that using an LM scaling factor value lower than that tuned for optimal ASR performance in terms of WER resulted in improved keyterm detection performance. This was seen to be true for both the P2 phase, in which lattices are generated using a bigram language

model, and the P3 phase where these lattices are re-decoded using a trigram language model. The scaling factors were reduced from 14.0 and 12.0 to 8.0 for the bigram and trigram language models respectively. The system lattices output from the final rescoring phase are those on which the STD experiments are carried out in this work. These lattices are converted from their romanised form to a normalised UTF-8 Arabic representation before the keyterm search is carried out entirely in this domain. As the underlying recogniser makes use of graphemes as a sub-word level representation, this is the sub-word unit representation utilised in the keyterm detection system.

The WER achieved by this recogniser varies between 62% and 81% across the eight communication links. The fact that the error rates are particularly high naturally has a significant impact on the level of difficulty in performing keyterm detection, as many keyterms could have been misrecognised. The RATS program for which the system was developed defines a list of 219 keyterms for evaluation, 64 of which are single-word terms with 155 multi-word terms. The keyterm detection operating point which is of particular interest in this program is that of a false alarm rate of 4%. All experiments in this work are based on the dev-1 dataset, which comprises data held out from the original training set for each of the communication channels. This dataset was further split into a dataset used to train the parameters of the CRF models, and a separate, held-out test set used for evaluation. The training subset contains 181 of the total 401 keyterm occurrences, with the remaining 220 occurrences being present in the test subset. Using the evaluation framework, the training data was scored such that a supervised training dataset could be created. The target labels for this system are effectively the labels indicating whether a hypothesised keyterm is a true positive or a false positive ("TP" or "FP"). A set of predictor features is associated with each such labelled keyterm hypothesis, which yields the training set utilised by the CRF models. The evaluation datasets have the same form, with the exception being that the target labels are absent. As is the case in the general confidence estimation framework, the confidence score assigned to each hypothesis is the marginal probability of the label corresponding to the hypothesis being correct, which is "TP" in the keyterm detection task.

9.3.1 Evaluation

For each keyterm detection hypothesised by the spoken term detection system developed and utilised in this work, a confidence score may be obtained using the CRF-based approach. The detection events,

along with the timings and scores are pooled from the system output for each keyterm across all segments of the recognised audio. The set of detections for each keyterm are then sorted in decreasing order by the confidence score. This data is output in the appropriate format for the NIST scoring tool utilised for evaluation in the RATS program. The output of this tool includes the number of keyterms which are detected (TP), missed (FN) and the number of false positives (FP). Another output of this evaluation tool is data which can be used to plot DET curves for the results, which will be the primary means of evaluating system performance in this work.

9.4 Experiments

The focus of the CRF-based keyterm confidence systems is that of the scores for hypothesised keyterms, and the level of performance that can be achieved at various operating points based on these scores. However, this performance is naturally related to the ability of the keyterm detection system to detect keyterms in the first instance, regardless of the scores assigned to these hypotheses. It is useful to analyse this performance in terms of the precision (P) and recall (R) of the system for this detection task. These metrics are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9.2}$$

where TP is the number of true positives (or "hits"), FP is the number of false positives and FN is the number of false negatives (or "misses"). Results in terms of the precision and recall achieved by both the word and grapheme-based component systems developed in this work are presented in Table 9.1 on the next page. These results are obtained when all keyterm detections are accepted (i.e. no score threshold is applied below which hypotheses are rejected).

As the word-based system generates few hypotheses, which are fairly likely, this system achieves much better precision than the grapheme-based system. However, the grapheme-based system does ultimately detect more of the keyterms present in the audio, resulting in a 46% relative improvement in recall over the word-based system. Although the precision of the grapheme-based system is very

	Train			Test		
System	TP	R	P	TP	R	P
Word	103	0.569	0.138	102	0.464	0.128
Grapheme	150	0.829	0.003	174	0.790	0.003

Table 9.1 *Results for the word and grapheme-based keyterm detection systems on training and test subsets of the Dev-1 data, with no threshold applied to confidence scores. TP = True Positives, R = Recall and P = Precision.*

low, this is desirable such that certain operating points may be reached. In addition, many spurious detections should be disregarded when a threshold is applied to the scores.

In the sections which follow, experiments in estimating keyterm confidence scores for the output of the word and grapheme-based components of the keyterm detection system will be presented, before those for the combined hybrid system are detailed.

9.4.1 *Word-based System: Keyterm Confidence*

Using the predictor features relevant for the word-based system, CRF models are trained to estimate improved confidence measures on this level. The DET curves in Figure 9.1 on the facing page show the results for various systems of this type.

It should be pointed out that the range of operating points that can possibly be reached is limited (the maximum false alarm (FA) rate is 0.09%). This is due to the fact that relatively few keyterm hypotheses, which are actually fairly likely to exist in the audio, are generated by this system. In all the CRF-based models, spline feature functions were applied to the continuous features, as these have been shown in this work to generally be the most effective representation of such predictor features. Five evenly-spaced knot points are used instead of the standard configuration which uses eight such knot points, as this configuration yields improved performance. This is a result of the fact that the data set is relatively small, and with a larger number of knot points the model does not generalise as well. The baseline system considered for these experiments is based on the unmapped word posteriors (LAPR). The confidence scores in this baseline system are calculated by taking the product of the posteriors (LAPR(K)) for each word comprising a keyterm. It can be seen from the DET curve that using the LAPR(K) feature in isolation with the CRF model yields performance similar to that of the baseline.

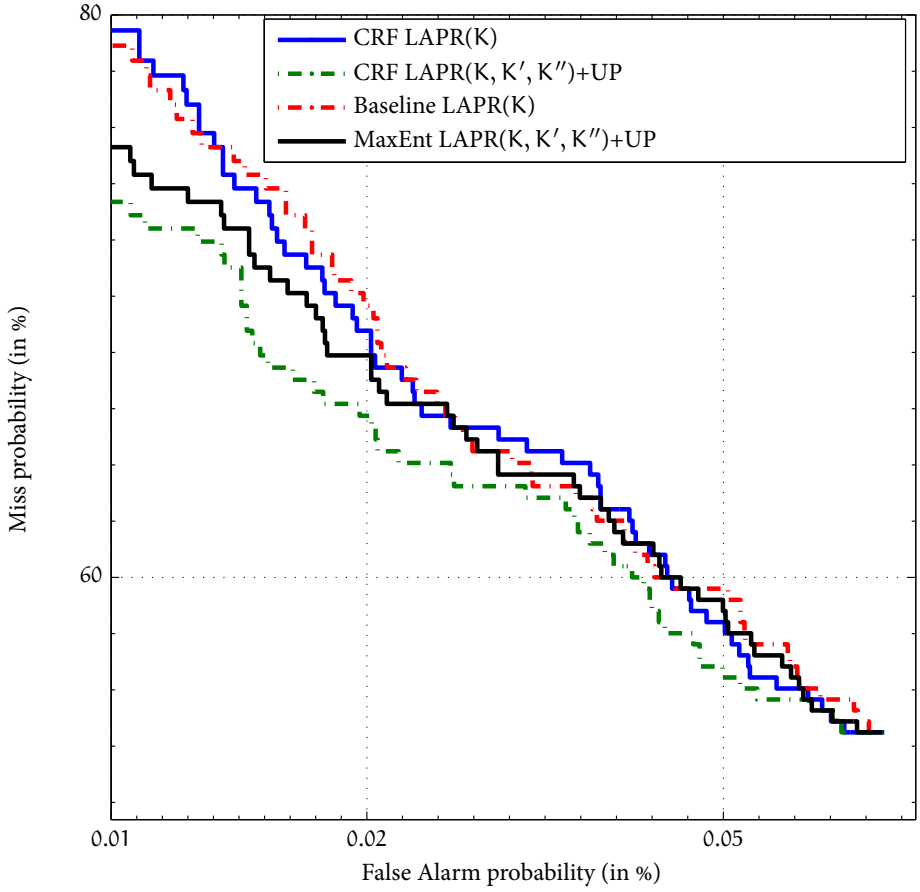


Figure 9.1 *DET curves showing performance of CRF-based confidence estimation applied in the word-level STD system.*

This result is to be expected, as no additional information is incorporated into the model which can effectively be used to improve the confidence scores. As predictor features are added to the models however, incremental performance improvements are achieved. The performance curves for each such incremental system are not shown in Figure 9.1 for the purposes of clarity, with attention being paid to systems which are useful in contrasts. The system which yielded the best performance is one in which the full set of predictor features are combined, this includes: the posterior of the keyterm LAPR(K), the unigram prior (UP) and the contextual posteriors (LAPR(K′) and LAPR(K″)). The DET curve for a model in which this configuration is used is also shown in Figure 9.1. In the CRF-based approach, each keyterm is represented as a sequence of separate words (with the associated predictor features

for each word). An alternative approach is considered in order to investigate whether the keyterms should indeed be represented in this manner, or as a single set of observations for the entire keyterm. To achieve this, the predictor features are averaged over the words of each keyterm in the data, thus yielding a single predictor feature (or set thereof) for each detection. Data of this form is subsequently used within an approach based on a non-sequential maximum entropy model. The results show that the CRF approach outperforms the equivalent non-sequential maximum entropy (MaxEnt) model which makes use of the same predictor features. It should be noted that the inclusion of the LITM1 feature functions does not improve performance in the word-based system. This is due to the fact that there are very few training examples for each keyterm, such that generalisation becomes an issue where these feature functions, and their parameters, are concerned.

9.4.2 *Sub-word-level System: Keyterm Confidence*

Experiments in improving the scores for detections in the grapheme-based system are presented in this section. Specific aspects related to this system are investigated through these experiments, such as the acoustic score-based predictor feature and direct score normalisation. Due to the productive nature of this component of the spoken term detection system, a significant number of detection hypotheses are generated. It is therefore possible to apply the literal moment (LITM1) feature functions for score normalisation here, as there is significant training data to reliably estimate the parameters for individual keyterms. The continuous feature related to the LITM1 feature functions (for which moment statistics are estimated) is taken to be the acoustic score ratio (LSAAR(K)) in these experiments. First-order moment parameters are therefore estimated for all keyterms which occur at least 100 times in the training data, of which there are 197. The list of these frequent keyterms will be referred to as the keyterm *shortlist*. An additional out-of-shortlist parameter is estimated to cover the remaining 22 infrequent keyterms. The results for grapheme-based keyterm detection experiments are presented in the DET plot of Figure 9.2 on the next page. It is immediately clear to see that these systems are able to reach a wider range of operating points than the word-based systems, as this DET plot extends beyond the 4% false alarm probability point on the x-axis. This is a consequence of the large number of keyterm hypotheses generated. More keyterms are therefore detected, with many more false alarms also being produced.

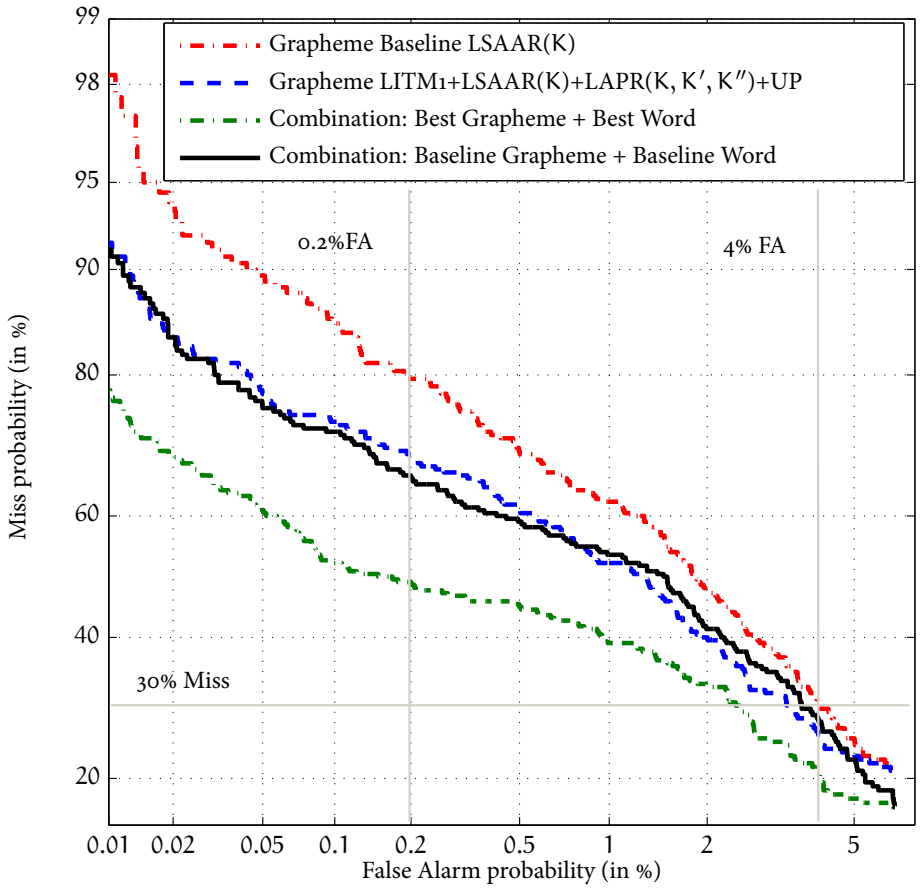


Figure 9.2 *DET curves showing performance of CRF-based confidence estimation in grapheme-based KWS, as well as in the combined word and grapheme-based KWS systems.*

The acoustic score ratio features (LSAAR(K)) are used as the confidence measures in the baseline system. This configuration is in line with that which might be employed in a simple keyterm detection system. Using this feature in isolation with the CRF model results in performance similar to the baseline, which is to be expected. However, incremental performance gains are achieved as additional useful features are included in the model. A configuration which extends the model through the use of LITM1 feature functions, and combines LSAAR(K), LAPR(K), contextual posterior features (LAPR(K′) and LAPR(K″)) and the unigram prior (UP) yields the best performance in the grapheme-based system. The DET curve for this system configuration is shown in Figure 9.2. This system shows relative improvements in miss probability over the baseline of 12.5% and 21.9% at false alarm rates

of 0.2% and 4% respectively. At a miss probability of 30% this system shows relative improvements in the false alarm rate of 14.6%. The contextual posteriors proved to be less useful in this system than in the word-based system. An explanation for this is that the graphemes used to form a keyterm hypothesis in the sub-word system are effectively allowed to have originally formed part of any parent word in the lattice. The confidence in the context of these parent words therefore has no real bearing on the quality of the keyterm hypothesis. The gains in performance evident in the figure for the best configuration are therefore primarily as a result of the inclusion of the LITM1 feature functions. The direct score normalisation approach proposed here is therefore shown to be particularly important, and useful, for this task.

9.4.3 Keyterm Confidence in the Hybrid System

Detection error trade-off (DET) curves showing the effect of combining the outputs of the word and grapheme-based systems in the hybrid keyterm detection configuration are shown along with the grapheme system curves in Figure 9.2 on the previous page. The baseline for these comparisons is formed by combining the individual word and grapheme-based system baselines. These baseline confidence scores are therefore not mapped before combination. However, it is interesting to note that after combination, this baseline performance is improved over a grapheme-based system used in isolation. This is a result of the additional true positives which are effectively gained from the inclusion of the word-based system, and come at a cost of relatively few false alarms. However, when systems for which confidence scores have been estimated using the approach presented in this work are combined, this results in considerable performance improvements over the individual systems and the baseline. The DET plot for the combined CRF system in Figure 9.2 on the preceding page corresponds to the combination of the best word and grapheme-based systems. It has already been shown that the individual performance of these systems is improved over their respective baselines to begin with. However, when these systems employing CRF-based confidence estimation are combined, further improvements are achieved. This is due to the fact that the confidence scores assigned to hypotheses within in each system are effectively normalised. The keyterm hypothesis scores generated by these different systems can thus be treated equally for decision making purposes. The performance of the word-based system is thereby maintained in the combined configuration, with the grapheme-based system contributing

many additional keyterm hypotheses, extending the operating range to desired levels. This results in a consistent relative improvement over the baseline of 26% in miss probability at false alarm rates of both 0.2% and 4%, as well as a relative improvement of 36% in the false alarm rate at a miss probability of 30%.

9.5 Summary Discussion

The approach proposed in this work, whereby the problem of assigning scores to keyterm detections is formulated as one of keyterm confidence estimation, is shown to improve the accuracy of these scores. This is achieved through the CRF-based combination of a number of novel discriminative predictor features introduced in this work. Including information in addition to the original posterior scores resulted in incremental performance gains in both the word and grapheme-based components of the system. This was true for both the contextual posteriors and the unigram priors in the word-based system. These features inform the model as to the quality of a hypothesis within its context, and the prior probability of the hypothesised keyterm in general, which resulted in improved score accuracy. It is shown that the CRF model is able to exploit the fact that a sequence of words comprises a keyterm, with this approach outperforming a competing non-sequential maximum entropy model.

An elegant approach to the problem of score normalisation across keywords, which makes use of keyterm-specific CRF feature functions to perform direct score normalisation is proposed as a contribution in this work. This approach is utilised to good effect within the grapheme-based system, in which sufficient data is available for estimating the parameters for these feature functions reliably. This development contributed the bulk of the performance gains observed in applying the keyterm confidence approach in the graphemic system. The use of the word posterior, unigram prior and the baseline acoustic posterior score does however yield incremental improvements in this system, thus supporting their use as additional information.

A further advantage of the CRF-based keyterm confidence approach, is seen when considering the considerable improvement in performance seen when the outputs of the individual component systems are combined. This is due to the effective inter-system normalisation which takes place, as the scores assigned in each of the separate systems are assigned using the principled CRF-based approach.

Chapter 10

Confidence in Deletions

The task of confidence estimation for ASR hypotheses is addressed in this thesis. The general scenario that has been assumed, is effectively that of performing binary classification. Where words hypothesised by an ASR system are classified as either being correct or incorrect. The definition of incorrect words in this sense is that of words which, as a result of a subsequent alignment with a reference hypothesis, may be deemed to either be *substitution* or *insertion* errors. These errors are defined relative to the reference. Substitution errors correspond to hypothesised words which are aligned with words in the reference, but do not match. Insertion errors correspond to words in the hypothesis which cannot be aligned with the reference.

Another type of error that may occur is that of *deletions*. These errors occur when there is a word absent from the ASR hypothesis. This implies that there is a word in the reference which cannot be aligned with a word in the hypothesis. This phenomenon, whereby words are missing from the ASR output, has not previously been addressed in the literature on confidence estimation to the authors knowledge.

In this chapter, an approach for directly modelling deletions as part of the confidence estimation process is described. This is a novel contribution made in this work. The proposed approach is firstly motivated from a conceptual standpoint. Thereafter, the structure and components of a CRF-based system for modelling deletion regions is detailed. Finally, the experimental framework and the results of a number of experiments investigating the accuracy with which this approach is able to model deletions are presented.

10.1 Deletion Regions

The direct approach to word-level confidence estimation taken in this work, makes use of a set of predictor features associated with a particular word. This information is used in order to estimate a measure of confidence in that word being correct. When considering the problem of modelling deletions, this paradigm is no longer valid. If a word is not hypothesised as part of the transcription output by an ASR system, then it is not clear where evidence for this hypothesis may be found within the system output. Consequently, no predictor features can be associated with a deleted word. This is irrespective of whether the underlying model or an alternative information source is used to compute features.

The concept which is therefore proposed here, is that of effectively modelling *deletion regions*. These regions correspond to intervals between words hypothesised by the ASR system that could potentially contain one or more deleted words. Given this definition, the approach is that of using information encoded in the sequence of hypothesised words and their associated predictor features, to predict when transitioning from an existing word hypothesis (which includes both correct and incorrect words) into a "missing" deleted word. The sequential nature of the CRF models applied to confidence estimation thus far in this thesis are particularly well suited to modelling this type of transition structure.

10.2 CRF models for Combined Confidence and Deletion Modelling

As was described in Section 10.1, the proposed approach uses a CRF to model so-called *deletion regions* in ASR output. In order to achieve this, it is assumed that a deletion can occur after any word. These unseen events must therefore be incorporated into the model in some way. This is achieved by augmenting the standard set of confidence estimation targets (indicating whether a word is correct or incorrect), with an indicator of whether the current word occurs before a deletion. This effectively doubles the size of the output label space $\mathcal{L}$. Separate observation feature functions are defined in relation to each output label in CRF models. This is desirable, as the characteristics of predictor features for words which occur before deletions are captured separately from those for words which are

not. The CRF also models transitions between the augmented labels, which should allow the model to capture characteristics of typical label sequences (e.g. multiple consecutive deletions being more or less frequent). In defining the augmented label set in this way, the model is capable of performing the standard confidence estimation task based on classifying words as being correct or incorrect, regardless of whether or not there is a deletion. The model is however also capable of learning something about the characteristics of deletions. This may include what conditions typically bring about deletions, as well as what potential impact this might have on the confidence scores for a particular word. The marginal probabilities corresponding to the pre-deletion and standard "Correct" labels are summed to yield the usual confidence score for a given word. Conversely, the marginal probabilities for the current word being both incorrect or correct, but occurring before a deletion, are summed to yield the deletion confidence measure.

As the approach defined in this way implies the model making a decision on whether a deletion should occur after the current time, the feature vector for each word is extended such that it consists of the predictor feature for not only the current word, but also the next word. This is intended to incorporate some sense of context around the placement of deletion boundaries. It should also compensate for the asymmetric nature of the task in predicting whether a deletion immediately follows a word, but not whether it could precede a word. It should be noted that for the last word in a sequence, the predictor features for the current word are copied into the space which would have been occupied by the features of the next word. This ensures the input predictor feature vector has a constant dimensionality. The special case in which a deletion region is present before the first word hypothesised in an utterance may also occur. To deal with such cases, the output label set is augmented with an indicator of this event having occurred. This results in a number of additional output labels for each of the base classes of labels. These additional labels effectively represent the end-deletion region boundary, rather than the standard begin-deletion region boundaries used in the more general case. For the purposes of this work, these two cases may be considered to both indicate a deletion, as one of the deletion boundaries is being detected in either case.

Figure 10.1 on the next page illustrates the proposed CRF-based approach for combined confidence and deletion modelling (or *deletion-informed confidence estimation*) with a synthetic training example. After aligning the reference and hypothesised sequences, it can be seen that there is a deletion error

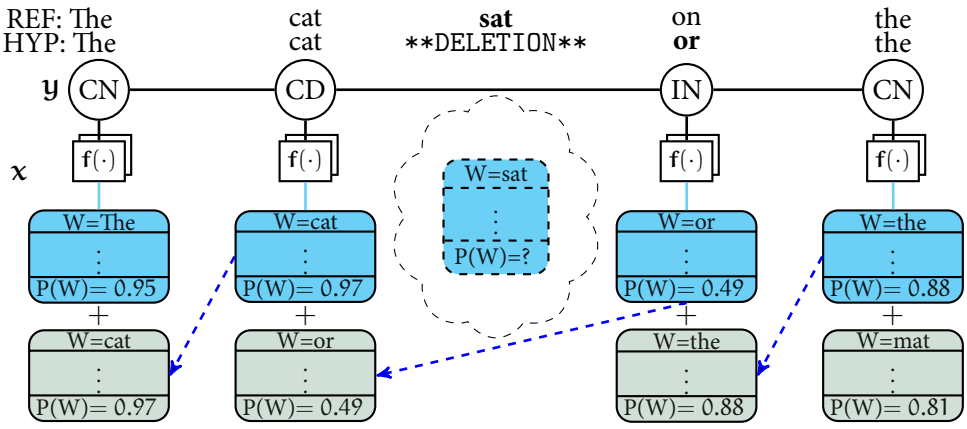


Figure 10.1 *Figure illustrating the proposed deletion-informed confidence estimation approach. A synthetic training example is shown, in which there is a deletion. The appropriate labels which indicate both whether the words are correct, as well as whether there is a deletion region following the current word are shown.*

corresponding to the word "sat". The standard confidence labels ("C" and "I") are augmented with a "D" or an "N" to indicate whether a deletion does or does not follow the current word. These labels form the sequence $\mathbf{y}$. The deletion region which occurs after the word "cat" (for which there are naturally no features) is also shown. The arrows depict how the current vector of predictor features ($\mathbf{x}$) is augmented with the right-context (i.e. the features for the next word). Feature functions $\mathbf{f}(\cdot)$, of the same types as those investigated for general confidence estimation (e.g. splines) are used to relate the feature vectors to the output labels ($\mathbf{y}$).

10.3 Evaluation

The techniques developed to model deletion regions are still capable of estimating confidence scores in the usual manner. Therefore, the confidence estimation aspect of these systems can be evaluated using those metrics used previously in this work, such as the word-level NCE metric. Evaluation in terms of this metric is in fact useful, as it is interesting to investigate whether accounting for deletions has a positive, negative, or neutral nett effect on the confidence estimation task in general.

As the task of having a confidence measure in deletions has not previously been studied (to the author's knowledge), there is no standard metric for evaluating performance for this task. Before such

a metric can be proposed in this work, the general task for evaluation should be defined. It is assumed that there is the potential for a deletion region to exist in a slot after every word hypothesised by an ASR system, as well as before the first word in every utterance. The goal in detecting deletions may therefore be formulated as being that of estimating a probability with which a deletion region is likely to occur in each such slot. This definition is analogous to that in the confidence estimation task for correct words. It is therefore sensible to use a modified version of the NCE metric for this purpose. The standard NCE metric is a normalised representation of the information gain in assigning confidence scores to the set of correct and incorrect words, over assuming the empirical accuracy of the system is the likelihood of each word being correct. For the deletion evaluation, the modified metric will represent the information gain in assigning scores to each of the potential deletion regions in the slots where these may occur, over assuming these scores are equal to the empirical deletion rate for the system. This results in the definition of the NCE(del) metric, defined as follows:

$$\mathrm{NCE}(\mathrm{del}) = \frac{\left(H_{\max}(\mathrm{del}) + \sum_{w\in\mathcal{D}} \log_2(\hat{p}_{\mathrm{del}}(w)) + \sum_{w\in\mathcal{N}} \log_2(1-\hat{p}_{\mathrm{del}}(w))\right)}{H_{\max}(\mathrm{del})} \tag{10.1}$$

where $\hat{p}_{\mathrm{del}}(w)$ is the confidence in a deletion occurring after a particular word w, $\mathcal{D}$ is the set of slots in which deletions occur and $\mathcal{N}$ is the set of slots in which deletions do not occur. Given d words which are considered as being deleted from the hypotheses (i.e. slots in which deletion regions occur), out of N total slots where deletions could occur, the empirical average probability of a deletion region being present is $P_d = \frac{d}{N}$, and

$$H_{\max}(\mathrm{del}) = -d\log_2(P_d) - (N-d)\log_2(1-P_d). \tag{10.2}$$

In addition to the NCE-based evaluation, the performance of a system for detecting deletions may be evaluated using DET curves, in which detection performance is measured over a range of operating points. These operating points are specified through the application of a thresholding decision applied to the scores. Detection performance is measured in terms of false alarm probability and miss probability. For deletion detection, misses correspond to deletions that are not detected and false alarms correspond to instances which are flagged as corresponding to deletion regions when there is no deletion in that slot.

10.4 Experiments

A number of experiments were carried out in which "deletion-informed" confidence estimation models are developed and evaluated. It is desired that these confidence estimation models are both capable of carrying out confidence estimation in the classical sense, whilst being able to simultaneously estimate the probability of deletions occurring after a given word.

The datasets for these experiments are essentially the same as those which were utilised for the word and sub-word-level confidence estimation tasks reported previously in this work. However, where previously the labels corresponding to each word or sub-word unit were based on whether they are correct or not, these labels must be augmented such that they also reflect when deletions may occur. In order to achieve this, the training data is scored in the usual manner using the NIST `SCLite` evaluation and scoring package. Using the output of this scoring tool, the desired labels can be assigned to each hypothesis in the training data. This set makes use of the standard confidence labels (`CORRECT` and `INCORRECT`) as a base, which are augmented with indicators for when deletion regions are present. This includes the standard deletion marker, which indicate that a deletion region follows the current word. In addition, the label for the special case which indicates when a deletion region occurs before the first word in the utterance are also produced. These special case labels are only ever observed for the first word of any sequence in training.

In keeping with the approach taken in the experiments for the classical confidence estimation approaches, a baseline which makes use of a decision tree with a subsequent piecewise linear mapping was developed using the training data for this task.

10.4.1 Word-level Deletion-informed Confidence Estimation

The premise of this approach is that the sequential nature of the CRF model can be used to learn when transitioning into a deletion region. In addition to this transition information, some predictor features related to the words which are actually hypothesised will undoubtedly have some relation to whether deletions are present in the transcription. For instance, words which have a particularly low lattice arc posterior ratio (LAPR - as defined in equation 7.4 on page 90), are likely to be within regions of low confidence. Within these regions it is more likely to encounter not only insertions and substitutions,

but also deletions.

In addition to word-level predictor features such as LAPR, it is put forward that sub-word level information may also contribute to the process. As an example, if the lattice sub-word arc posterior ratio (LSAPR) for the sub-words of a word effectively "tail off" and decrease towards the end of the word, this may also indicate increased uncertainty in the output. This may also increase the likelihood of a deletion occurring immediately thereafter, as the recogniser is uncertain of its hypotheses. This effect may also be evident when considering the posteriors generated by an alternative sub-word recogniser (as investigated in Section 8.1). The posterior probabilities output by such a system are not constrained by any language model, and may therefore yield sharper changes in sub-word scores when the hypothesised word is more unlikely. The identity of the hypothesised sub-word units is suspected to be useful for detecting deletions. In particular, if the presence of silence (`sil`) is hypothesised on the sub-word-level at the end of a word, this may indicate a situation where there is some audio which is modelled as silence when it shouldn't be. This would result in no word being output for the corresponding time span, potentially causing a deletion error.

Experiments in which CRF-based models are trained to take advantage of characteristics of both the sequential aspect of transitioning into deletion regions, whilst also making use of a set of predictor features to contribute to this process are presented in this section. These models assign scores to every word which indicates whether the current word is correct or incorrect, as well as whether it occurs on the onset of a deletion region (resulting in a deletion confidence score). For the deletion detection aspect, these systems are evaluated in terms of the "deletion" NCE expressed in equation 10.1, using the deletion confidence scores. The standard NCE scores for confidence estimation are also quoted, and are computed based on the standard confidence scores output by the systems. The results for these experiments are presented in Table 10.1 on the next page. As the DNCE is a not a standard metric computed by the NIST evaluation software, a tool was developed to evaluate the system output and compute both the DNCE and NCE metrics. The resulting NCE scores quoted here show slight discrepancies with the word-level NCE scores quoted for the standard confidence estimation experiments, as a result of some of the expansions and replacements that are made by the NIST scoring tool. For the purposes of these experiments, this discrepancy is not critical, as the standard NCE metric is not the primary means of evaluation for deletion-informed confidence estimation.

	dev10d		eval09ns	
System	DNCE	NCE	DNCE	NCE
DT Baseline LAPR	-0.047	0.314	-0.019	0.358
MaxEnt LAPR	-0.044	0.312	0.010	0.351
① LSAPR	0.098	0.248	0.093	0.232
② ASWP	0.034	0.047	0.042	0.056
③ LAPR	0.117	0.334	0.128	0.356
④ ASWP+LAPR	0.127	0.347	0.133	0.361
⑤ LSAPR+ASWP	0.101	0.250	0.096	0.232
⑥ POST (M1)	0.118	0.316	0.134	0.332
⑦ POST	0.130	0.347	0.137	0.361
⑧ POST+POSTCON (M1)	0.121	0.320	0.136	0.331
⑨ POST+POSTCON	0.134	0.347	0.143	0.362
⑩ POST+POSTCON+SIL	0.134	0.347	0.144	0.362
⑪ DELONLY POST+POSTCON+SIL	0.110	–	0.133	–
⑫ DELONLY POST+POSTCON	0.109	–	0.132	–

Table 10.1 *Results for systems capable of detecting deletions, evaluated in terms of the deletion metric (DNCE), and the standard confidence metric (NCE) - where applicable. Results are shown for the dev10d and eval09ns datasets. Unless otherwise indicated, spline feature functions are applied to continuous predictor features. POST=LAPR+LSAPR+ASWP. POSTCON=LAPR(t+1)+LSAPR(t+1)+ASWP(t+1). DELONLY is used to depict systems for deletion detection only.*

In Table 10.1, the decision tree baseline is in fact made up of two systems, one trained specifically for the classical confidence estimation task and one for the deletion region detection task. As is seen from the results, NCE scores for the standard confidence estimation task are in line with previous word-level results (see Section 7.4.1). However, the performance in terms of the deletion metric (DNCE) is poor, with the scores being negative. This indicates that this system in fact performs worse than a naïve system which assumes the empirical deletion rate of the system is equal to the probability of a deletion occurring after each word. This is to be expected, as there is not necessarily a high degree of correlation between the current word-level posterior used by this model (LAPR), and the likelihood of a deletion following this word. This fact, coupled with the inability of the model to capture the sequence

information which is vital in this task therefore yields a poor modelling approach. This result is verified by making use of a non-sequential maximum entropy model in which the LAPR feature is represented using spline feature functions. This system yields similar performance to the baseline, proving that the sequence information is crucial. Approaches which attempt to classify the onset of deletion regions based purely on the observations for a current word are therefore clearly not able to perform this task.

Comparing the CRF-based system (3) which makes use of the word-level posterior only (LAPR), with that of the baseline or the maximum entropy model using the same information, it is immediately clear that the sequential nature of the model contributes significantly to the modelling approach. This sequential structure is exploited to make it possible for deletion regions to be modelled and thereby detected. The results in terms of the standard NCE metric for this system show that the inclusion of the deletion modelling aspect does not have a negative impact on the model's capability to generally assign confidence scores (in terms of whether the words are correct). As was observed previously in this work, the CRF-based system outperforms the decision tree baseline on the dev10d dataset, and is roughly equivalent on the eval09ns dataset. It should be noted that an evenly-placed spline configuration is employed by this model for ease of comparison with other systems. This approach was shown previously to have marginally worse performance on eval09ns. It is expected performance would be improved using the automatic spline placement approach (this was shown in the results of Section 7.4.2).

The set of posterior predictor features denoted as "POST" considered for some of the experiments shown in Table 10.1 on the preceding page include the lattice arc posterior ratio (LAPR), the lattice sub-arc posterior ratio (LSAPR) and the alternative recogniser sub-word posterior (ASWP).

Results for a CRF-based system which makes use of the full set of posterior predictor features by applying the spline feature functions to them are shown (system 7). Large improvements over the system making use of LAPR in isolation (system 3) are observed, both in terms of evaluation for the classical confidence estimation task (NCE) and in the deletion task (DNCE). This improvement is 11.1% and 7% relative on dev10d and eval09ns respectively, and 3.9% and 1.4% relative in NCE on dev10d and eval09ns respectively. These results highlight the utility of these predictor features for both tasks.

The word hypothesised by the system immediately after the current word in a sequence may be immediately following a deletion region, and the predictor features for this word may be indicative of

this situation to some degree. These predictor features for the next hypothesised word shall be referred to as the *right-context* features. It is considered useful for the system to be able to make local decisions on whether there is a deletion following the current word, based on information pertaining to both the current word, and this right-context information. The predictor feature vector for a hypothesised word is therefore augmented with the right-context predictor features. The results for systems 8,9 and 10 in Table 10.1 on page 190 are based on this configuration, which is denoted as "POSTCON". Applying spline feature functions to these predictor features in addition to the other posterior features (system 9) yields relative improvements in DNCE over a system without the right-context features of 3.1% and 4.4% on dev10d and eval09ns respectively. These improvements are not as large as may have been expected. This may be attributed to the fact that the right-context for the final word in a sequence doesn't exist. The dimensions of the predictor features which normally correspond to the additional context are taken to be that of the current (i.e. final) word in this case. This fact may dilute the discriminative power of this contextual information.

In order to verify previous results which showed that splines outperformed other feature functions for continuous representations, a configuration which applies first order moment statistics to the full set of predictor features is evaluated (system 8). The results for this system re-iterate those obtained in previous experiments (see Section 7.4.1). Here, the spline-based systems outperform the first order moment-based system by 10% and 5% relative in DNCE on dev10d and eval09ns respectively.

A system which is intended to capture potential effects caused by the presence of silence in the recogniser output is also investigated (system 10 in Table 10.1 on page 190). This is achieved by including an input predictor feature which is essentially a (discrete) string value that indicates whether there is a silence in the sub-word sequence hypothesised by the system for the current or right-context word. The string match (SM) feature functions detailed in Section 8.3.2 are applied to this predictor feature. Extending the CRF-based system using these feature functions results in marginal improvements in performance for both the deletion and classical confidence estimation tasks on datasets. This small improvement may be attributed to the fact that the presence of silence is also associated with non-deletion labels by the model. The model may therefore not be able to distinguish between what is effectively silence corresponding to non-deletions, and that which corresponds to the onset of a deletion region. Furthermore, there are significantly more datapoints where silence is present for non-deletion events

than there are for deletion events in the training data. The parameters of the model may therefore be slightly skewed to the more generic non-deletion case, which is encountered more often.

The results reported for this task show that the performance of the confidence estimation task is not affected negatively by the additional capability of the model to detect deletions. In order to investigate whether the converse is true, systems were trained with labels indicating the presence of deletions only, and not whether the word is correct or incorrect. These models are therefore purely deletion detection models. The results for these systems are shown in the bottom two lines of Table 10.1 on page 190 and correspond to systems 11 and 12. The first interesting point to consider is that this system yields considerably inferior DNCE performance to that of the combined model. This implies that the information encoded in whether or not a word is correct is very useful in predicting the likelihood of deletions occurring. Furthermore, a system which combines the posterior predictor features with the silence information once again yields minimal improvements. A similar trend was observed in the deletion-informed confidence estimation system. In that system, this is attributed to the fact that any information related to the presence of silence is lost when the model considers the presence of this indicator in relation to confidence labels as well as deletion labels. The fact that this result is also observed in the deletion-only system serves to further underline the inefficacy of this feature as a discriminative feature in predicting deletions. It is therefore concluded that there is no real signal in this predictor feature which can be exploited to improve deletion detection performance, both in considering deletion-informed confidence estimation and deletion detection systems.

10.4.1.1 Analysis of Detection Error Trade-off Curves

The performance of the deletion-informed confidence estimation systems at detecting deletions in ASR output is also evaluated in terms of DET curves. This provides insight into the performance achieved by these systems over a wide range of operating points. These operating points are set by applying a threshold to the deletion detection score. The DET curves for a selection of contrasting systems are shown in Figure 10.2 on the next page for the combination of the eval09ns and dev10d datasets. The data is combined in plotting the DET curves as the curves for the individual datasets are quite similar in terms of comparing the relative performance of these systems.

In the DET curves of Figure 10.2 on the following page, the random performance line is shown as

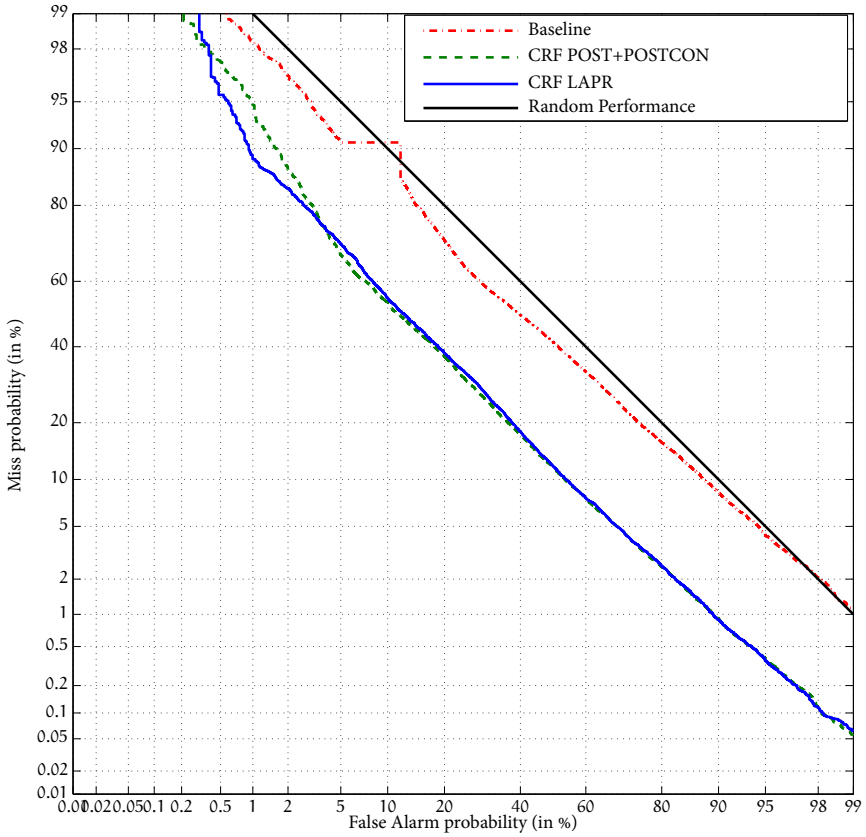


Figure 10.2 *DET curves showing the performance of CRF-based deletion-informed confidence estimation systems in terms of deletion detection on the combination of the eval09ns1 and dev10d datasets. Curves are shown for the baseline system, a CRF with the current set of posterior features ("POST") and a system using these posteriors as well as the right-context "POSTCON" features. The random performance line is also shown on this plot.*

a point of comparison. Curves below this line perform better than random, and are therefore deemed useful for the detection task. The baseline systems is shown to perform better than random over almost all operating points, although the margin between the two diminishes with increasing false alarm probabilities. This shows that while the non-sequential baseline approach does not yield positive DNCE scores, the system does capture some aspects of the problem, as it is able to perform better than random.

The CRF-based system which makes use of the LAPR predictor feature in isolation performs well, and yields large improvements across all operating points. For instance, the relative improvements in miss probabilities at false alarm probabilities of 5% and 95% are 91.7% and 22.7%. The relative

improvements in false alarm probabilities at miss probabilities of 5% and 95% are 28.04% and 83.3% respectively. These large improvements echo the large gains observed in the DNCE-based evaluation of this system. The system which includes the full set of posterior features for this task, as well as those corresponding to the right-context, does not show improvements over the single-feature CRF system between operating points corresponding to 0.4% and 3.5% false alarm probabilities/rates. Seemingly, at these relatively low false alarm probabilities, the LAPR predictor feature is a good estimator for deletions. This makes sense as low posterior probabilities (and correspondingly high deletion scores) may be within general regions of low confidence where deletions are more likely to occur. The right-context information may be less useful in these situations. However, at other operating points where the deletion scores are not as high, the full-featured system is shown to outperform the single-feature system (although by a small margin) over a larger range of operating points, as it is able to leverage some of the contextual information. These results support those observed in the DNCE-based evaluation.

The DET curves show that the modelling approaches taken for the deletion detection task are indeed capable of addressing the problem to some extent, as they outperform a non-sequential baseline considerably. In addition, the fact that the baseline system shows improved performance over a random assignment shows that the choice of this baseline is reasonable.

Chapter 11

Conclusion

The theme of this thesis has been that of improving confidence estimation for speech recognition hypotheses. A powerful, flexible framework based on a sophisticated machine learning approach (conditional random fields) was developed from the ground up, extended, and refined. This framework was applied to classical confidence estimation scenarios, on both the word and sub-word levels. The problem of assigning scores to keyterms detected in a spoken term detection system was formulated as a more general confidence estimation task. This made it possible for the confidence estimation framework to be applied as a principled approach to this problem. The sequential nature of the CRF-based approach was exploited so that a previously unexplored area of research could be investigated, which aims to simultaneously detect deletions in ASR output, and estimate confidence scores in hypothesised words.

In the sections which follow, conclusions will be drawn relating to each of the contributions made and the research questions investigated in this thesis.

11.1 Word-level Confidence Estimation

The CRF-based framework developed in this work was applied to the word-level confidence estimation task. This resulted in significant overall improvements in performance, yielding a state-of-the art confidence estimation system. The sequential nature of the model, which captures the transition structure of successive correct or incorrect words is a contributing factor to these improvements. This is evid-

enced by the fact that the CRF-based approach was shown to outperform comparable non-sequential maximum entropy models. The problem of representing continuous predictor features was investigated, with spline feature functions proving to be the most useful technique. A variant of these feature functions in which spline knot points are placed in a data-driven manner was proposed, and shown to yield improved representations of the continuous predictor features. The differences in performance achieved using the various representations for continuous input features show that it is an important consideration. A procedure for extracting predictor features from lattices for both 1-Best hypotheses, and alternative hypotheses was proposed. This technique made it possible for a number predictor features introduced in this work to be extracted from the recognition lattices for both types of hypotheses. Predictor features which are posterior probabilities of various forms, and are extracted from the lattice (e.g. lattice arc posterior ratio (LAPR), lattice arc language model posterior (LALMP)), were shown to be the most discriminative when incorporated into the CRF-based confidence estimation model. A method to compute measures of acoustic stability "on the fly" without requiring multiple decoding passes was proposed. The application of this technique yields the lattice acoustic stability (LAS) and lattice posterior stability (LPS) predictor features which were in the second most discriminative set of predictor features investigated. Incorporating these novel predictor features into the model resulted in improvements in confidence estimation performance, with the full-featured system yielding considerable gains in NCE relative to a strong baseline of up to 11.4%. These results were also shown to be significant to a level of below 0.1%.

Confidence estimation as carried out in this work, is proposed as a means through which longer-range information can be incorporated into a speech recognition system. As recognisers cannot feasibly account for such information themselves, including this information as part of the confidence estimation may result in improved processing of the ASR output. The TF*IDF measure, typically used in the information retrieval domain, was used for this task and was shown to contribute to the confidence estimation process. This was particularly true for the larger evaluation dataset, as the estimated values for this measure were effectively more reliable.

The problem of obtaining confidence scores for hypotheses other than the lattice 1-Best was addressed through the use of novel techniques. Projecting the alternative hypothesis onto a lattice, and subsequently extracting predictor features using the generalised approach described in this work was

shown to be useful. Moreover, it was seen that the lattice-based posterior (LAPR) predictor feature obtained in this manner, is a more accurate indicator of confidence in isolation than the confusion network posteriors. These confusion network posteriors are available when the hypothesis annotated with confidence scores is the output of a confusion network. The hypothesis injection/projection approach is effectively one in which alternative information is incorporated into the confidence estimation framework. Such alternative information was shown to be the most useful when combining multiple predictor features to improve confidence score estimates. In general, larger gains were observed when combining additional predictor features from other sources, rather than when including more features from the underlying source of the hypothesis for which scores are estimated.

It was observed that in using a sequential model for confidence estimation, the tendency of errors to occur consecutively in sequences is exploited to yield improvements. In a linear-chain CRF, this sequence structure is limited, as it is effectively a first-order chain. Extensions were made to the CRF framework developed in this work, whereby hidden states can be incorporated into the models. The desired effect of doing so being to model longer sequences of correct or incorrect words, using the hidden structure at a level higher than that of the word level. In the novel application of hidden-state models of this type to the word-level confidence estimation task, considerable improvements were achieved, even within the fully-featured systems. The full-featured hidden-state system showed relative performance improvements over a strong baseline of up to 17.2%. In fact, these gains were much larger than those typically observed when including additional predictor features, and are amongst some of the largest improvements reported in this work.

To a certain extent, the results of experiments carried out in this work suggest there is something of a ceiling to performance improvements that can be achieved in adding additional predictor features from similar sources of information. This was addressed by adding predictor features extracted based on other information, such as the alternative recogniser scores and the TF*IDF measures. Another means through which improvements are achieved above this ceiling is by improving the statistical modelling of the features, such as with the hidden-state approach.

11.2 Sub-word-level Confidence Estimation

The use of sub-word-level information was put forward as a potential alternative source of information to that typically available on the word-level. Such information was subsequently used in estimating confidence scores for both words and sub-words. The combined benefits of having information defined at this level, with that of incorporating an additional alternative source of information into the confidence estimation process were explored. This was achieved using an alternative sub-word-level recogniser for this purpose, with the predictor features extracted from it proving to be of great use.

In the direct sub-word-level confidence estimation task, which is defined purely on the sub-word level, incremental improvements are seen with the inclusion of each sub-word predictor feature in the confidence estimation system. Predictor features of this type are therefore seen to contribute the most to the performance improvements achieved. However, the addition of word-level information was also useful, and resulted in further gains. This is in lieu of the fact that the task is actually defined solely on the sub-word level. The results for the full-featured sub-word-level system showed the utility of the sub-word-level approach, with large performance improvements in NSWCE of up to 31.7% relative to the baseline.

The concept that sequences of consecutive correct or incorrect sub-words tend to occur was further exploited on this level, to yield some of the largest improvements in performance reported in this work. This was achieved through the application of hidden-state CRF models to this task. The hidden state structure captures the dynamics of these longer-span sequences of consecutive sub-words. As modelling is carried out on the sub-word-level, the number of units over which these regions may range is potentially longer than on the word-level, and may begin or end within words. The model captures these characteristics effectively and exploits them in yielding more accurate confidence score estimates. A full-featured sub-word-level CRF system yields relative improvements in NSWCE performance of up to 48.6% over the baseline system. This is an important result, as it serves to highlight the fact that modelling improvements are capable of yielding large gains for confidence estimation. It also shows that there is hidden structure in the sub-word-level sequence which can be leveraged for confidence estimation.

In using a model defined at the sub-word-level for estimating word-level confidence scores, the task presents somewhat different challenges. These challenges are a result of the mismatch between

the level of modelling and the level of granularity at which the output is required. The primary issue being that of the loss of word-level structure, such that the model is not able to treat successive sub-words from different words differently. A method was proposed whereby this could be accounted for by engineering feature functions to capture this word-level transition information in addition to the sub-word-level transition structure. The use of the resulting word boundary feature functions provided sub-word-level systems which yielded results equivalent to word-level approaches, when considering only word-level predictor features. The sub-word-level approach did however prove useful when incorporating sub-word-level predictor features into the system. With the sub-word-level system acting directly on predictor features defined on this level, it is capable of exploiting this information more effectively than equivalent word-level systems using averaged representations of these features. When the full set of word-level predictor features is incorporated into the system for this word-level task, the contribution of the sub-word-level predictor features is diminished. Consequently, the benefits of modelling on this level become less prominent. The full-featured sub-word system therefore yielded NCE performance equivalent to the word-level system for the fully-featured configuration, with relative improvements of up to 11.4% over the baseline. Such an approach should however be considered in some cases. For instance, when only sub-word-level predictor features are available, this approach should be taken in favour of averaging the input to the word level. This would also true when the most informative predictive features are expected to be those defined on the sub-word level.

11.3 Keyterm Confidence

The problem of assigning scores to detections of keyterms in a spoken term detection system was approached as a more general confidence estimation problem. The application of CRF models to this form of discriminative score mapping was therefore proposed and investigated in this work. A hybrid (word/sub-word) keyterm detection system was also developed. This was necessary so that predictor features, such as the novel contextual posteriors and the lattice sub-arc acoustic ratio (LSAAR), could be extracted during the keyterm search process. The problem of score normalisation across detections for different keyterms was addressed in both the word and sub-word components of the hybrid system. The unigram prior predictor features which indicate how likely keyterms are in general text/language,

were particularly useful in improving performance for the word-level system. In the sub-word level system, where more data is available, a method of accounting for score normalisation directly within the model was proposed. In this approach, keyterm-specific parameters are estimated for the distribution of the LSAAR predictor feature. This effectively provides a mechanism whereby the model is able to penalise or boost scores on a term-by-term basis, such that the output scores are different from what it might have estimated using only the generic distribution information for LSAAR over all keyterms. Improvements in performance were achieved within the individual word and sub-word systems, with the largest improvements being achieved in the sub-word-level graphemic system. This is partly due to the fact that there is more data on which the systems can be trained, but is also a characteristic of the system in general. As there are a large number of hypotheses generated in the sub-word-based system, the accuracy of the scores is even more important. Spurious hypotheses should be part of a long tail of the score distribution, such that they do not feature in the majority of operating points. Interestingly, the LAPR predictor feature for the words from which sub-words are taken to produce complete keyterm hypotheses proved useful. The use of this predictor feature yielded decent improvements in performance. This effect is attributed to two factors, the first being that these scores encode language model information not present in the acoustic-only posteriors (LSAAR). The second factor is that these predictor features are effectively additional measures of quality for the sub-word-level hypotheses. The general likelihood of the words indicate how likely a particular concatenation of sub-word units is likely to be, based on their respective parent words. In combining the word and graphemic systems, large improvements in performance are achieved over the baseline systems which do not employ CRF-based score mapping. Here, relative improvements of 26% and 36% relative are seen at the operating points of interest. This is partly due to the general improvements and score normalisation within each system. The combined system improvements are also attributed to the fact that keyterm scores are effectively normalised across systems by way of the CRF mapping applied in each component, before their outputs are combined. Applying a single global threshold to the scores in the combined system is therefore sensible with the scores being normalised in this way.

The large improvements achieved in these systems serve to show that a principled confidence-based approach to estimating scores for keyterm hypotheses is indeed very useful. The direct model-based technique for score normalisation provides an elegant solution to this problem, which contributes to

these improvements. Although the approach necessitates that features be extracted during search, it is not a particularly intensive process (compared with the search). The improvements achieved in taking this approach therefore far outweigh this relatively small cost.

11.4 Deletion Detection

The ability to detect deletions at all is an entirely novel contribution of this work. Detecting the presence of deletions in ASR output was cast as an extension of the confidence estimation problem. The proposed approach exploits the sequential nature of the CRF model. This is possible when combined with the concept of deletion regions of one or more deleted words which may occur between words. It was shown that taking this approach results in a system which is capable of simultaneously performing confidence estimation in a general sense, and estimating a measure indicating whether a deletion region is likely to occur following a word in the ASR system hypothesis. The results showed that the key component of this approach is indeed the sequential nature of the problem. Non-sequential systems such as the decision tree baseline and maximum entropy model were shown to effectively be unable to perform this task successfully. It was observed that the word level posterior (LAPR) proved useful in predicting deletions. The sub-word-level posterior predictor features also contribute meaningfully in combination to yield incremental improvements in the accuracy with which the model is capable of detecting deletions. In addition, the ability of the model to estimate accurate confidence scores does not suffer, and is in fact improved slightly. This result shows that the two tasks are closely related. The approach proposed in this work whereby these tasks are combined in "deletion-informed" confidence estimation is therefore shown to be a sensible one. The use of predictor features which indicate the presence of silence was expected to be informative, but resulted in little or no improvements in performance. It was therefore concluded that there is no real signal in the presence or absence of silence, as this occurs frequently in the context of non-deletions events as well as for deletions.

11.5 Future work

The use of hidden states within the CRF model was seen to be capable of capturing additional aspects of the sequence structure in the confidence estimation problem, at a level different from that at which modelling takes place. The proposed approach for detecting deletions within ASR output exploits the sequential structure of linear-chain CRF models to capture the characteristics of when a deletion is likely to occur after a given word. The standard constraints applied for hidden states in this work assert that the value of this variable must increase from left to right within successive output labels of the same type, and reset to zero when transitioning to a new output label. These constraints are not sensible for application in the deletion detection task. This is a result of the fact that the output labels will change on the boundaries of deletion regions, such that the hidden state values will reset and remain zero on both beginning and ending boundaries for deletions, as is imposed by the hidden-state constraints. However, if different constraints are applied specifically for this task, it may be possible to capture information pertaining to regions in which there is general uncertainty due to the presence of one or more deletion regions. The application of hidden-state CRF models to deletion-informed confidence estimation would result in an interesting, principled approach, which warrants investigation.

The idea of incorporating long-range information into the confidence estimation process is one which is particularly interesting. The lack of long-range context is often put forward as a deficiency of speech recognition systems. However, incorporating such information into these systems is challenging, and is largely infeasible. The framework for confidence estimation developed here provides a principled method through which this information may be leveraged to further enhance the output of the recognition system. The experiments in this work investigated the use of a single predictor feature of this type, the TF*IDF score. This information proved useful. It is therefore expected that other predictor features or sources of long-range information may also contribute to the confidence estimation process. The development and use of additional approaches for including long-range information within the confidence estimation process therefore warrants future investigation.

Recently, there has been a resurgence in the use of neural networks, and so-called *deep* neural networks. This is particularly true in the fields of computer vision and speech recognition. These discriminative models make use of a large number of stacked hidden layers within a neural network. One of the factors often cited as being the reason these approaches yield improvements for acoustic

modelling, is that the deep structure of the network effectively induces features from the data, and learns the interrelation between these features in higher layers. Some of the improvements achieved in this work are based on feature function engineering, which is used to address certain known aspects of the problem. These are effectively hand-crafted features. The combination of this "expert" feature engineering aspect, with the ability of a model to effectively induce features relating to potentially unknown characteristics of the problem from the data automatically, should yield a powerful modelling approach. In work related to deep architectures, conditional neural fields (Peng *et al.* 2009; Do and Artières 2010) have been proposed as a model which augments the standard CRF model with non-linear units similar to the neurons present in neural networks. Extending the CRF toolkit developed in this work (`CRFTK`) such that similar layers of non-linear units may be included in the CRF models is an interesting area in which the work of this thesis could be developed further. This may also be extended further such that hidden-state CRF models which include such non-linear feature extraction layers may be investigated.

Bibliography

C. Allauzen, M. Mohri, and M. Saraclar (2004). "General indexation of weighted automata: application to spoken utterance retrieval." In *Proceedings of the HLT-NAACL 2004 Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*.

T. Anastasakos and S. V. Balakrishnan (1998). "The use of confidence measures in unsupervised adaptation of speech recognizers." In *Proceedings of ICSLP*.

L. Bahl, P. Brown, P. V. De Souza, and R. Mercer (1986). "Maximum mutual information estimation of hidden Markov model parameters for speech recognition." In *Proceedings of ICASSP*.

J. Besag (1974). "Spatial Interaction and the Statistical Analysis of Lattice Systems." *Journal of the Royal Statistical Society. Series B (Methodological)* 36 (2), pp. 192–236.

G. W. Blackwood, A. de Gispert, and W. Byrne (2010). "Fluency constraints for minimum bayes-risk decoding of statistical machine translation lattices." In *Proceedings of COLING*.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing (2004). "Confidence estimation for machine translation." In *Proceedings of COLING*.

H. Bourlard and C. Wellekens (1990). "Links between Markov models and multilayer perceptrons." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (12), pp. 1167–1178.

J. S. Bridle (1973). "An efficient elastic-template method for detecting given words in running speech." In *Proceedings of the British Acoustic Society Meeting*.

M. G. Brown, J. T. Foote, G. J. F. Jones, K. S. Jones, and S. J. Young (1996). "Open-vocabulary speech indexing for voice and video mail retrieval." In *Proceedings of ICM*.

D. Can and M. Saraclar (2011). "Lattice indexing for spoken term detection." *IEEE Transactions on Audio, Speech, and Language Processing* 19 (8), pp. 2338–2347.

L. L. Chase (1997*a*). *Error-responsive feedback mechanisms for speech recognizers*. Ph.D. thesis, Carnegie Mellon University.

L. Chase (1997*b*). "Word and acoustic confidence annotation for large vocabulary speech recognition." In *Proceedings of Eurospeech*.

C. Chelba and A. Acero (2005). "Position specific posterior lattices for indexing speech." In *Proceedings of ACL*.

R. Christiansen and C. K. Rushforth (1977). "Detecting and locating key words in continuous speech using linear predictive coding." *IEEE Transactions on Acoustics, Speech and Signal Processing* 25 (5), pp. 361–367.

S. Cox and R. C. Rose (1996). "Confidence measures for the switchboard database." In *Proceedings of ICASSP*.

S. Cox and S. Dasmahapatra (2002). "High-level approaches to confidence estimation in speech recognition." *IEEE Transactions on Speech and Audio Processing* 10 (7), pp. 460–471.

S. Davis and P. Mermelstein (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." *IEEE Transactions on Acoustics, Speech and Signal Processing* 28 (4), pp. 357–366.

F. Diehl, M. J. F. Gales, X. Liu, M. Tomalin, and P. C. Woodland (2011). "Word boundary modelling and full covariance gaussians for Arabic speech-to-text systems." In *Proceedings of Interspeech*.

T.-M.-T. Do and T. Artières (2010). "Neural conditional random fields." In *Proceedings of AISTATS*.

R. O. Duda, P. E. Hart, and D. G. Stork (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

J. P. Egan (1975). *Signal detection theory and ROC-analysis*. Series in cognition and perception. Academic Press.

G. Evermann and P. C. Woodland (2000*a*). "Posterior probability decoding, confidence estimation and system combination." In *Proceedings of NIST Speech Transcription Workshop*.

G. Evermann and P. C. Woodland (2000*b*). "Large vocabulary decoding and confidence estimation using word posterior probabilities." In *Proceedings of ICASSP*.

J. Fayolle, F. Moreau, C. Raymond, G. Gravier, and P. Gros (2010). "CRF-based combination of contextual features to improve a posteriori word-level confidence measures." In *Proceedings of Interspeech*.

M. J. F. Gales (1998). "Maximum likelihood linear transformations for HMM-based speech recognition." *Computer Speech and Language* 12 (2), pp. 75–98.

M. J. F. Gales and P. C. Woodland (1996). "Mean and variance adaptation within the MLLR framework." *Computer Speech and Language* 10, pp. 249–264.

M. J. F. Gales and S. Young (2007). "The application of hidden Markov models in speech recognition." *Foundations and Trends in Signal Processing* 1 (3), pp. 195–304.

M. J. F. Gales and F. Flego (2012). "Model-based approaches for degraded channel modelling in robust ASR." In *Proceedings of Interspeech*.

J. Garofolo, C. Auzanne, and E. Voorhees (2000). "The TREC spoken document retrieval track: A success story." In *Proceedings of TREC*.

M. T. Gibson (2008). *Minimum Bayes risk acoustic model estimation and adaptation*. Ph.D. thesis, University of Sheffield.

L. Gillick and S. J. Cox (1989). "Some statistical issues in the comparison of speech recognition algorithms." In *Proceddings of ICASSP*.

L. Gillick, Y. Ito, and J. Young (1997). "A probabilistic approach to confidence estimation and evaluation." In *Proceedings of ICASSP*.

A. Gispert, G. Blackwood, G. Iglesias, and W. Byrne (2013). "N-gram posterior probability confidence measures for statistical machine translation: an empirical study." *Machine Translation* 27 (2), pp. 85–114.

V. Goel and W. J. Byrne (2000). "Minimum Bayes-risk automatic speech recognition." *Computer Speech and Language* 14 (2), pp. 115–135.

V. Goel, S. Kumar, and W. Byrne (2001). "Confidence based lattice segmentation and minimum bayes-risk decoding." In *Proceedings of Eurospeech*.

J. Goodman (2004). "Exponential priors for maximum entropy models." In *Proceedings of HLT-NAACL*.

D. Graff, S. Sessa, S. Strassel, and K. Walker (2011). "RATS Data Plan." Tech. rep., Linguistic Data Consortium.

A. Gunawardana, H.-W. Hon, and L. Jiang (1998). "Word-based acoustic confidence measures for large-vocabulary speech recognition." In *Proceedings of ICSLP*.

A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt (2005). "Hidden conditional random fields for phone classification." In *Proceedings of Interspeech*.

J. M. Hammersley and P. Clifford (1971). "Markov field on finite graphs and lattices."

J. H. L. Hansen, R. Huang, B. Zhou, M. Seadle, J. Deller, J. R., A. R. Gurijala, M. Kurimo, and P. Angkititrakul (2005). "SpeechFind: advances in spoken document retrieval for a national gallery of the spoken word." *IEEE Transactions on Speech and Audio Processing* 13 (5), pp. 712–730.

T. J. Hazen, T. Burianek, J. Polifroni, and S. Seneff (2002). "Recognition confidence scoring for use in speech understanding systems." *Computer Speech and Language* 16 (1), pp. 49–67.

G. Heigold, H. Ney, R. Schluter, and S. Wiesler (2012). "Discriminative Training for Automatic Speech Recognition: Modeling, Criteria, Optimization, Implementation, and Performance." *IEEE Signal Processing Magazine* 29 (6), pp. 58–69.

H. Hermansky (1990). "Perceptual linear predictive (PLP) analysis of speech." *The Journal of the Acoustical Society of America* 87 (4), pp. 1738–1752.

G. Hernandez-Abrego and J. Marino (2000). "Contextual confidence measures for continuous speech recognition." In *Proceedings of ICASSP*.

Y. Hifny and S. Renals (2009). "Speech recognition using augmented conditional random fields." *IEEE Transactions on Audio, Speech and Language Processing* 17 (2), pp. 354–365.

A. Higgins and R. Wohlford (1985). "Keyword recognition using template concatenation." In *Proceedings of ICASSP*.

D. A. James (1996). "A system for unrestricted topic retrieval from radio news broadcasts." In *Proceedings of ICASSP*.

H. Jiang (2005). "Confidence measures for speech recognition: A survey." *Speech Communication* 45 (4), pp. 455–470.

D. Johnson (2004). "ICSI Quicknet software package." ⟨http://www1.icsi.berkeley.edu/Speech/qn.html⟩.

B.-H. Juang, W. Hou, and C.-H. Lee (1997). "Minimum classification error rate methods for speech recognition." *IEEE Transactions on Speech and Audio Processing* 5 (3), pp. 257–265.

S. O. Kamppari and T. J. Hazen (2000). "Word and phone level acoustic confidence scoring." In *Proceedings of ICASSP*.

S. Katz (1987). "Estimation of probabilities from sparse data for the language model component of a speech recognizer." *IEEE Transactions on Acoustics, Speech and Signal Processing* 35 (3), pp. 400–401.

T. Kemp and T. Schaaf (1997). "Estimating confidence using word lattices." In *Proceedings of Eurospeech*.

R. Kneser and H. Ney (1995). "Improved backing-off for M-gram language modeling." In *Proceedings of ICASSP*.

A. Krogh and S. Riis (1999). "Hidden neural networks." *Neural Computation* 11 (2), pp. 541–563.

N. Kumar (1997). *Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition*. Ph.D. thesis, John Hopkins University.

J. Lafferty, A. McCallum, and F. Pereira (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." In *Proceedings of ICML*.

C.-H. Lee, J.-L. Gauvain, R. Pieraccini, and L. R. Rabiner (1993). "Large vocabulary speech recognition using subword units." *Speech Communication* 13 (3-4), pp. 263–279.

C. J. Leggetter and P. C. Woodland (1995). "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models." *Computer Speech and Language* 9 (2), pp. 171–185.

H. Liao and M. J. F. Gales (2005). "Joint uncertainty decoding for noise robust speech recognition." In *Proceedings of Interspeech*.

D. C. Liu and J. Nocedal (1989). "On the limited memory BFGS method for large scale optimization." *Mathematical Programming* 45 (1-3), pp. 503–528.

X. Liu, M. Gales, and P. C. Woodland (2003). "Automatic complexity control for HLDA systems." In *Proceedings of ICASSP*.

K. Livescu, E. Fosler-Lussier, and F. Metze (2012). "Subword modeling for automatic speech recognition: past, present, and emerging approaches." *IEEE Signal Process. Mag.* 29 (6), pp. 44–57.

E. Lleida and R. C. Rose (1996). "Likelihood ratio decoding and confidence measures for continuous speech recognition." In *Proceedings of ICSLP*.

E. Lleida, J. B. Mariño, J. M. Salavedra, A. Bonafonte, E. Monte, and A. Martinez (1993). "Out-of-vocabulary word modelling and rejection for keyword spotting." In *Proceedings of Eurospeech*.

W. K. Lo, F. K. Soong, and S. Nakamura (2004). "Generalized posterior probability for minimizing verification errors at subword, word and sentence levels." In *Proceedings of ISCSLP*.

B. Logan, P. Moreno, and O. Deshmukh (2002). "Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio." In *Proceedings of HLT*.

B. Logan, P. Moreno, J.-M. V. Thong, E. Whittaker, J. manuel Van, and T. Whittaker (1996). "An Experimental Study Of An Audio Indexing System For The Web." In *Proceedings of ICSLP*.

J. Mamou, B. Ramabhadran, and O. Siohan (2007). "Vocabulary independent spoken term detection." In *Proceedings of SIGIR*.

L. Mangu, E. Brill, and A. Stolcke (1999). "Finding consensus among words: Lattice-based word error minimization." In *Proceedings of Eurospeech*.

L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon (2013). "Exploiting diversity for spoken term detection." In *Proceedings of ICASSP*.

L. Mangu, E. Brill, and A. Stolcke (2000). "Finding consensus in speech recognition: word error minimization and other applications of confusion networks." *Computer Speech and Language* 14 (4), pp. 373–400.

A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, and M. A. Przybocki (1997). "The DET curve in assessment of detection task performance." In *Proceedings of Eurospeech*.

A. McCallum, D. Freitag, and F. Pereira (2000). "Maximum entropy Markov models for information extraction and segmentation." In *Proceedings of ICML*.

E. McDermott, T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri (2007). "Discriminative training for large-vocabulary speech recognition using minimum classification error." *IEEE Transactions on Audio, Speech, and Language Processing* 15 (1), pp. 203–223.

D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish (2007). "Rapid and accurate spoken term detection." In *Proceedings of Interspeech*.

M. Mohri (1996). "On some applications of finite-state automata theory to natural language processing." *Natural Language Engineering* 2 (1), pp. 61–80.

L. P. Morency, A. Quattoni, and T. Darrell (2007). "Latent-dynamic discriminative models for continuous gesture recognition." In *Proceedings of CVPR*.

P. J. Moreno, B. Logan, and B. Raj (2001). "A boosting approach for confidence scoring." In *Proceedings of Interspeech*.

P. Motlícek (2009). "Automatic out-of-language detection based on confidence measures derived from LVCSR word and phone lattices." In *Proceedings of Interspeech*.

P. Motlicek, F. Valente, I. Szoke, P. Motlícek, F. Valente, I. Szöke, P. Motlicek, F. Valente, and I. Szoke (2012). "Improving acoustic based keyword spotting using LVCSR lattices." In *Proceedings of ICASSP*.

A. Nadas (1983). "A decision theorectic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 31 (4), pp. 814–817.

C. Neti, S. Roukos, and E. Eide (1997). "Word-based confidence measures as a guide for stack search in speech recognition." In *Proceedings of ICASSP*.

H. Ney, U. Essen, and R. Kneser (1994). "On structuring probabilistic dependencies in stochastic language modelling." *Computer Speech and Language* 8, pp. 1–38.

J. Neyman and E. S. Pearson (1933). "On the problem of the most efficient tests of statistical hypotheses." *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231 (694-706), pp. 289–337.

NIST (2006). "The spoken term detection (STD) 2006 evaluation plan." ⟨http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf⟩.

J. J. Odell (1995). *The use of context in large vocabulary speech recognition*. Ph.D. thesis, University of Cambridge.

J. Olive, C. Christianson, and J. McCary (2011). *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. SpringerLink : Bücher. Springer.

Z. Ou and H. Luo (2012). "CRF-based confidence measures of recognized candidates for lattice-based audio indexing." In *Proceedings of ICASSP*.

M. Padmanabhan, G. Saon, and G. Zweig (2000). "Lattice-Based Unsupervised MLLR For Speaker Adaptation." In *Proceedings of the ISCA workshop on adaptation methods for speech recognition*.

Y.-C. Pan, H. lin Chang, B. Chen, and L.-S. Lee (2007). "Subword-based position specific posterior lattices (s-PSPL) for indexing speech information." In *Proceedings of Interspeech*.

S. Parlak and M. Saraclar (2008). "Spoken term detection for Turkish Broadcast News." In *Proceedings of ICASSP*.

J. Peng, L. Bo, and J. Xu (2009). "Conditional neural fields." In *Proceedings of NIPS*.

J. Pinto and R. N. V. Sitaram (2005). "Confidence measures in speech recognition based on probability distribution of likelihoods." In *Proceedings of Interspeech*.

L. C. W. Pols (1977). *Spectral analysis and identification of Dutch vowels in monosyllabic words*. Free University, Amsterdam.

D. Povey and P. C. Woodland (2002). "Minimum Phone Error and I-smoothing for improved discriminative training." In *Proceedings of ICASSP*.

A. Quattoni, M. Collins, and T. Darrell (2004). "Conditional random fields for object recognition." In *Proceedings of NIPS*.

M. G. Rahim (1997). "Discriminative utterance verification for connected digits recognition." *IEEE Transactions on Speech and Audio Processing* 5 (3), pp. 266–277.

M. G. Rahim, C.-H. Lee, and B.-H. Juang (1995). "Robust utterance verification for connected digits recognition." In *Proceedings of ICASSP*.

J. R. Rohlicek, W. Russell, S. Roukos, H. Gish, J. R. Rohliceb, and H. Gidh (1989). "Continuous hidden Markov modeling for speaker-independent word spotting." In *Proceedings of ICASSP*.

R. C. Rose, B. H. Juang, and C. H. Lee (1995). "A training procedure for verifying string hypotheses in continuous speech recognition." In *Proceedings of ICASSP*.

R. C. Rose and D. B. Paul (1990). "A hidden Markov model based keyword recognition system." In *Proceedings of ICASSP*.

B. Rueber (1997). "Obtaining confidence measures from sentence probabilities." In *Proceedings of Eurospeech*.

H. Sakoe and S. Chiba (1978). "Dynamic programming algorithm optimization for spoken word recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1), pp. 43–49.

A. Sanchis, A. Juan, and E. Vidal (2003). "Estimating confidence measures for speech recognition verification using a smoothed naive Bayes model." In *Proceedings of IbPRIA*.

A. Sanchis, A. Juan, and E. Vidal (2012). "A word-based naıve Bayes classifier for confidence estimation in speech recognition." *IEEE transactions on Audio, Speech and Language Processing*. 20 (2), pp. 565–574.

M. Saraclar and R. Sproat (2004). "Lattice-based search for spoken utterance retrieval." In *Proceedings of HLT-NAACL*.

M. S. Seigel and P. C. Woodland (2011). "Combining information sources for confidence estimation with CRF models." In *Proceedings of Interspeech*.

M. S. Seigel and P. C. Woodland (2012). "Using sub-word-level information for confidence estimation with conditional random field models." In *Proceedings of Interspeech*.

M. S. Seigel, P. C. Woodland, and M. J. Gales (2013). "A confidence-based approach for improving keyword hypothesis scores." In *Proceedings of ICASSP*.

A. R. Setlur, R. A. Sukkar, and M. G. Rahim (1996). "Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training." In *Proceedings of ICASSP*.

M. Siu and H. Gish (1999). "Evaluation of word confidence for speech recognition systems." *Computer Speech and Language* 13 (4), pp. 299–319.

M. Siu, H. Gish, and F. Richardson (1997). "Improved estimation, evaluation and applications of confidence measures for speech recognition." In *Proceedings of Eurospeech*.

K. Spärck Jones (1972). "A statistical interpretation of term specificity and its application in retrieval." *Journal of Documentation* 28 (1), pp. 11–21.

Y.-H. Sung and D. Jurafsky (2009). "Hidden conditional random fields for phone recognition." In *Proceedings of ASRU*.

C. Sutton and A. McCallum (2004). "Collective segmentation and labeling of distant entities in information extraction." In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.

C. Sutton and A. Mccallum (2006). *Introduction to conditional random fields for relational learning*, MIT Press.

J. Tejedor, D. T. Toledano, M. Bautista, S. King, D. Wang, and J. Colás (2010). "Augmented set of features for confidence estimation in spoken term detection." In *Proceedings of Interspeech*.

K. Thambiratnam and S. Sridharan (2007). "Rapid yet accurate speech indexing using dynamic match lattice spotting." *IEEE Transactions on Audio, Speech and Language Processing* 15 (1), pp. 346–357.

M. Tomalin, F. Diehl, M. J. F. Gales, J. Park, and P. C. Woodland (2010). "Recent improvements to the Cambridge Arabic Speech-to-Text systems." In *Proceedings of ICASSP*. pp. 4382–4385.

V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young (1997). "MMIE training of large vocabulary recognition systems." *Speech Communication* 22 (4), pp. 303 – 314.

D. Vergyri, I. Shafran, A. Stolcke, V. R. R. Gadde, M. Akbacak, B. Roark, and W. Wang (2006). "The SRI/OGI 2006 spoken term detection system." In *Proceedings of Interspeech*.

H. Wallach (2003). "Efficient training of conditional random fields." In *Proceedings of 6th Annual CLUK Research Colloquium*.

F. Wallhoff, D. Willett, and G. Rigoll (2000). "Frame-discriminative and confidence-driven adaptation for LVCSR." In *Proceedings of ICASSP*.

D. Wang, S. King, J. Frankel, and P. Bell (2009). "Term-dependent confidence for out-of-vocabulary term detection." In *Proceedings of Interspeech*.

D. Wang, J. Tejedor, S. King, and J. Frankel (2012). "Term-dependent confidence normalisation for out-of-vocabulary spoken term detection." *Journal of Computer Science Technolgy* 27 (2), pp. 358–375.

S. B. Wang, A. Quattoni, L. P. Morency, D. Demirdjian, and T. Darrell (2006). "Hidden conditional random fields for gesture recognition." In *Proceedings of CVPR*.

M. Weintraub (1995). "LVCSR log-likelihood ratio scoring for keyword spotting." In *Proceedings of ICASSP*.

M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke (1997). "Neural-network based measures of confidence for word recognition." In *Proceedings of ICASSP*.

F. Wessel, K. Macherey, and R. Schluter (1998). "Using word probabilities as confidence measures." In *Proceedings of ICASSP*.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney (2001). "Confidence measures for large vocabulary continuous speech recognition." *IEEE Transactions on Speech and Audio Processing* 9, pp. 288–298.

C. White, J. Droppo, A. Acero, and J. Odell (2007). "Maximum entropy confidence estimation for speech recognition." In *Proceedings of ICASSP*.

D. Willett, A. Worm, C. Neukirchen, and G. Rigoll (1998). "Confidence measures for HMM-based speech recognition." In *Proceedings of ICSLP*.

G. Williams and S. Renals (1997). "Confidence measures for hybrid HMM/ANN speech recognition." In *Proceedings of Eurospeech*. Rhodes, pp. 1955–1958.

G. Williams and S. Renals (1999). "Confidence measures from local posterior probability estimates." *Computer Speech and Language* 13, pp. 395–411.

J. G. Wilpon, C. H. Lee, and L. Rabiner (1989). "Application of hidden Markov models for recognition of a limited set of words in unconstrained speech." In *Proceedings of ICASSP*.

J. G. Wilpon, L. Rabiner, C.-H. Lee, and E. R. Goldman (1990). "Automatic recognition of keywords in unconstrained speech using hidden Markov models." *IEEE Transactions on Acoustics, Speech and Signal Processing* 38 (11), pp. 1870–1878.

M. J. Witbrock and A. G. Hauptmann (1997). "Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents." In *Proceedings of ICDL*.

P. C. Woodland (2001). "Speaker adaptation for continuous density HMMs: a review." In *Proceedings of the ISCA workshop on adaptation methods for speech recognition*.

P. C. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones (2000). "Effects of out of vocabulary words in spoken document retrieval." In *Proceedings of SIGIR*.

P. C. Woodland and D. Povey (2002). "Large scale discriminative training of hidden Markov models for speech recognition." *Computer Speech and Language* 16 (1), pp. 25 – 47.

H. Xu, D. Povey, L. Mangu, and J. Zhu (2010). "An improved consensus-like method for Minimum Bayes Risk decoding and lattice combination." In *Proceedings of ICASSP*.

S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland (2006). *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK.

S. J. Young, N. H. Russell, and J. H. S. Thornton (1989). "Token passing: a simple conceptual model for connected speech recognition systems." Tech. rep., Cambridge University Engineering Department.

S. J. Young and P. C. Woodland (1994). "State clustering in HMM-based continuous speech recognition." *Computer Speech and Language* 8 (4), pp. 369–394.

S. R. Young (1994). "Detecting misrecognitions and out-of-vocabulary words." In *Proceedings of ICASSP*.

S. J. Young, J. Odell, and P. Woodland (1994). "Tree-based state tying for high accuracy acoustic modelling." In *Proceedings of HLT*.

D. Yu, L. Deng, and A. Acero (2009). "Using continuous features in the maximum entropy model." *Pattern Recognition Letters* 30 (14), pp. 1295–1300.

D. Yu, S. Wang, J. Li, and L. Deng (2010). "Word confidence calibration esing a maximum entropy model with constraints on confidence and word distributions." In *Proceedings of ICASSP*.

P. Yu, K. Chen, C. Ma, and F. Seide (2005). "Vocabulary-independent indexing of spontaneous speech." *IEEE Transactions on Speech and Audio Processing* 13 (5-1), pp. 635–643.

T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel (1997). "Recognition of conversational telephone speech using the JANUS speech engine." In *Proceedings of ICASSP*.

T. Zeppenfeld and A. H. Waibel (1992). "A hybrid neural network, dynamic programming word spotter." In *Proceedings ICASSP*.

B. Zhang, R. Schwartz, S. Tsakalidis, L. Nguyen, and S. Matsoukas (2012). "White listing and score normalization for keyword spotting of noisy speech." In *Proceedings of Interspeech*.

R. Zhang and A. I. Rudnicky (2001). "Word level confidence annotation using combinations of features." In *Proceedings of Interspeech*.

J. Zheng and A. Stolcke (2005). "Improved discriminative training using phone lattices." In *Proceedings of Interspeech*.