Out-of-Vocabulary Spoken Term Detection

Dong Wang

University of Edinburgh

Doctor of Philosophy

Institute for Communicating and Collaborative Systems

School of Informatics

University of Edinburgh

2010

Abstract

Spoken term detection (STD) is a fundamental task for multimedia information retrieval. A major challenge faced by an STD system is the serious performance reduction when detecting out-of-vocabulary (OOV) terms. The difficulties arise not only from the absence of pronunciations for such terms in the system dictionaries, but from intrinsic uncertainty in pronunciations, significant diversity in term properties and a high degree of weakness in acoustic and language modelling.

To tackle the OOV issue, we first applied the joint-multigram model to predict pronunciations for OOV terms in a stochastic way. Based on this, we propose a stochastic pronunciation model that considers all possible pronunciations for OOV terms so that the high pronunciation uncertainty is compensated for.

Furthermore, to deal with the diversity in term properties, we propose a term-dependent discriminative decision strategy, which employs discriminative models to integrate multiple informative factors and confidence measures into a classification probability, which gives rise to minimum decision cost.

In addition, to address the weakness in acoustic and language modelling, we propose a direct posterior confidence measure which replaces the generative models with a discriminative model, such as a multi-layer perceptron (MLP), to obtain a robust confidence for OOV term detection.

With these novel techniques, the STD performance on OOV terms was improved substantially and significantly in our experiments set on meeting speech data.

Acknowledgements

There are too many people I owe my gratitude when this thesis finally reaches its function. Particularly, utmost thanks to my supervisor Simon King. It is utterly impossible to finish this thesis without Simon's skillful supervision: his deep insight to science, global perspective in research and wide rang of knowledge ensured my research always going to the correct direction and encountered problems being solved in an efficient way. I also owe much to his great help on study and life, from English correction to train ticket booking. Having the chance to work with such a gentleman is really a lucky pleasure.

Next I need thank my second supervisor, Joe Frankel, from whom I got lots of the ideas materialised in this thesis. Working with Joe is always relaxing and inspiring: he has such an ability to reach the essence of problems quickly and then raise his novel thoughts. Thank you Joe, for the invaluable help on research, and the delicious Christmas dinner.

A huge thank to the Marie Curie fellowship that supported my research and living expense in Edinburgh, very generously; thanks to the very kind project manager, Maria Wolters, who was always there whenever I needed help. Thanks also go to my third supervisor, Jim Scobbie, for the valuable instruction at the beginning of my study; Igor from BUT, for sharing the search tool; and Javi, Ivan and Peter, for the great effect on collaborative work. Thank my thesis examiners, Honza and Hiroshi, for the interest and time.

Thanks to Steve, you make CSTR so inspiring in research and friendly in life. Thanks to Rob, Hiroshi, Korin, Junichi, Matthew, Mike, the talks with you were full of pleasure. Thanks to all the folks at CSTR, especially my officemates, Ravi, Sebas, Tanja, Martin, Joao, you always made the PhD joyful and exciting.

A big thank you to my Chinese friends, Xingkun, Zhang Le, Songfang, Yansong, it is really my pleasure to meet you in Edinburgh. Thank you Avril, Alison, Jenny, Caroline, your logistic support made the study and life much easier.

Finally, I owe much to my families for whom I should have done more. Thanks for your understanding, my parents and my wife; thanks for your babbling, Wang Chun and Jingjing; thanks for your poems and songs, Xiao Mo.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Dong Wang*
University of Edinburgh)

To Ling, my wife, from whom I always get support when struggling with troubles.

Table of Contents

List of Figures

List of Tables

Symbols

O	speech segment
Q	phoneme sequence
q	phoneme
$\tilde{q}$	phoneme segment
G	grapheme sequence
g	grapheme
$\tilde{g}$	grapheme segment
U	graphone sequence
u	graphone
W	word sequence
K	search term
d	detection
$c(d)$	confidence measure of a detection d
$c_{lattice}$	lattice-based confidence
c_{disc}	discriminative confidence
c_{pron}	pronunciation confidence
c_{match}	soft match confidence
c_{mlp}	MLP-based direct posterior confidence
t	time

Acronyms

AM	Acoustic Model
ASR	Automatic Speech Recognition
ATWV	Average Term-Weighted Value
BNEWS	Broadcast News
CART	Classification And Regression Tree
CTS	Conversational Telephone Speech
CONMTG	Conferences & Meetings
DET	Detection Error Tradeoff
FA	False Alarm
HMM	Hidden Markov Model
IHM	Individual Headset Microphone
INV	In-Vocabulary
LM	Language Model
LTS	Letter-To-Sound
LVCSR	Large Vocabulary Continuous Speech Recognition
MDM	Multiple Distant Microphone
MLP	Multiple Layer Perceptron
OOV	Out-Of-Vocabulary

PER	Phoneme Error Rate
SDR	Spoken Document Retrieval
SPM	Stochastic Pronunciation Model
STD	Spoken Term Detection
SVM	Support Vector Machine
WER	Word Error Rate

Chapter 1

Introduction

1.1 Information retrieval from speech

Information is crucially important for modern societies. People rely on various information to make decisions, foster new ideas, arrange social activities, or just make entertainment. However, information is not friendly in nature: desired information is usually blemished by noise and cluttered with trifling details. How to retrieve requested information from various sources reliably and efficiently has become an important subject of research.

Information retrieval (IR), at least for text documents, has achieved remarkable progress. Success in research has fostered success in business, e.g., *google*, the famous information service provider, has grown at a tremendous speed in recent years. Nevertheless, retrieving information from other media still remains a hard problem: for example, from speech. Compared to text, speech is a more natural way for people to share knowledge and exchange ideas, and usually contains more information. Manually retrieving this information is extremely costly, if not impossible, as nobody wants to listen to a long audio file from the beginning just to find an interesting word or sentence. Therefore, an automatic retrieval method is highly desirable for retrieving information from speech.

A field of research has grown up around 'speech-based information retrieval', including key word spotting, topic discovery, automatic summarisation, spoken document retrieval (SDR), spoken data mining (SDM), etc. This thesis focuses on a fundamental task: spoken term detection (STD), which aims to retrieve spoken terms from a large volume of speech archives reliably and efficiently. An STD system provides a *google-like* search engine for speech; moreover, it lays a foundation for high-level

applications, such as SDR and SDM.

1.2 Spoken term detection (STD)

1.2.1 Task definition

Spoken term detection, or STD, was defined by National Institute of Standards and Technology (NIST) with the aim to *search vast, heterogeneous audio archives for occurrences of spoken terms*. To encourage research and development of this key technology, NIST organises an open evaluation series on STD. The first pilot evaluation took place in 2006 [NIST, 2006] on three conditions: broadcast news (BNEWS), conversational telephone speech (CTS) and conferences & meetings (CONFMTG). Three languages were involved in this evaluation: English, Arabic and Mandarin.

Figure 1.1 illustrates the standard framework of an STD system. In this framework, speech signals are first transcribed by a *speech recogniser* to a certain form of intermediate representation, e.g., word/subword transcripts or lattices; and then a *term detector* searches the intermediate representation to find putative occurrences of the terms in search; finally a *decision maker* judges each putative occurrence and determines if it is a reliable detection or a mistake. We call the speech recogniser in this architecture the *ASR subsystem*, and the term detector and the decision maker the *STD subsystem*.

In STD, the input query is a short word sequence, which is called a *search term*. An actual instance of a search term might be successfully detected or could be missed by the system; on the other hand, a hypothesised existence of a search term might be either a correct detection or a mistake. In this thesis, we call a real existence of a search term an *occurrence*, and a hypothesised existence found by the term detector a *detection*. If a detection corresponds to an occurrence, we call it a *hit*, otherwise it is a *false alarm* (FA). An occurrence that is failed to be detected is called a *miss*.

1.2.2 Relation of STD to other tasks

From the standard architecture, we see that STD relies on automatic speech recognition (ASR); in that sense, it belongs to the family of ASR research. ASR has been developed for more than a half century, and has achieved significant success with respect to accuracy and efficiency [Waibel and Lee, 1990; Huang et al., 2001]. The basic ASR task is speech transcription which has evolved into large vocabulary continuous

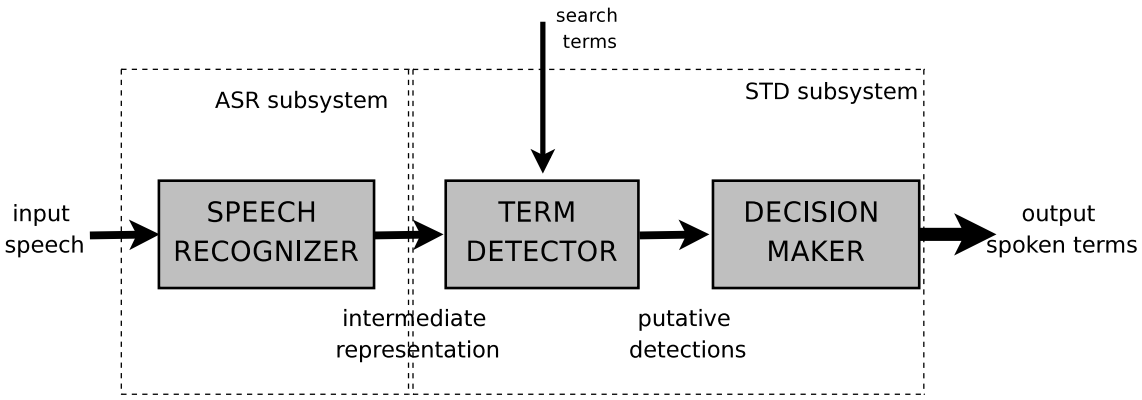

Figure 1.1: The standard STD architecture. There are three components in this architecture: a speech recogniser used to transcribe speech to intermediate representations (e.g., phoneme lattices); a term detector used to search intermediate representations to find putative occurrences of terms in the query; and a decision maker used to assert reliable detections and reject false alarms. An STD system works in two phase: at the offline indexing phase, speech documents are transcribed and archived; at the online detection phase, queries in the form of short word sequences are searched for within the archives and potential occurrences of the queries are found. An example of the STD output can be found in [Wang, 2009].

speech recognition (LVCSR) and has achieved rather high accuracy on read speech and is being extended to spontaneous speech. Besides transcription, some other tasks are also based on ASR technology, including speech activity detection (SAD), audio alignment, voice quality scoring, content summarisation, and STD. In this section, we compare STD with some other ASR tasks that are most relevant, including speech transcription, keyword spotting and SDR.

Relation of STD to speech transcription Speech transcription is the core task of ASR and the foundation of STD. According to the standard framework in Figure 1.1, an STD system uses a speech transcriber to convert input speech to intermediate representations, which means that the accuracy of the transcription directly affects the performance of the whole STD; on the other hand, the STD performance is not wholly *determined* by the accuracy of the transcription, since they have different objectives: for transcription, the target is a low word error rate (WER), so all words are treated equally; for STD, however, only the search terms are important. Therefore, to obtain a good performance for STD, an ideal speech transcriber in the ASR subsystem would be accurate on search terms but insensitive to other words, such as ‘the’, ‘that’, ‘their’

and other function words.

Relation of STD to keyword spotting The goal of keyword spotting and STD are similar: both aim at detecting interesting words or word sequences from speech. The main difference lies in that traditional keyword spotting detects keywords from audio streams while STD detects search terms from intermediate representations; however this is not absolute, as more and more keyword spotting systems use transcripts or lattices as intermediate representations as well. Another difference is that a keyword spotting system usually has a fixed vocabulary while an STD system has an *open* vocabulary, which means that an STD system is able to search for any term without re-transcribing the speech. Again, this is not absolute, since many researchers present open-vocabulary keyword spotting systems as well. The final difference is in the evaluation metrics: STD uses some new evaluation metrics defined by NIST, particularly the average term-weighted value (ATWV), whereas keyword spotting uses conventional metrics such as the figure of merit (FOM). In our mind, therefore, STD is just a 'modern name' of keyword spotting, following the standard architecture and the standard evaluation metrics defined by NIST.

Relation of STD to spoken document retrieval SDR and STD are similar in that they both aim at retrieving some interesting segments from speech, so they some retrieval techniques, e.g., lattice search, subword unit modelling, confidence estimation, etc. The difference is that SDR is not interested in the exact positions of occurrences of particular terms, instead it retrieves entire speech segments. Therefore, SDR is less affected by the transcription accuracy than STD, as it can use some high-level information, e.g., document word frequency, long span language models, semantic structures, etc. On the other hand, STD is one major approach to SDR [Jones et al., 1996a; Itoh et al., 2006] besides the LVCSR-based and subword ASR-based approaches [Schäuble and Wechsler, 1995; Witbrock and Hauptmann, 1997; Ng, 1998; Srinivasan and Petkovic, 2000; Cardillo et al., 2002; Saraclar and Sproat, 2004; Ma and Li, 2005].

1.3 Current research on STD

As we have mentioned, although STD is a newly defined task, the basic technologies and methodologies have been developed for a long time in related research areas: the

ASR techniques required for the ASR subsystem have been studied for 50 years; the retrieval techniques required for the STD subsystem have been developed and applied to keyword spotting and SDR successfully. For STD research, most studies assume a standard ASR subsystem with a normal level of accuracy and focus on the STD subsystem, designing an intelligent detection approach that can effectively use the imperfect transcription (typically 30% WER on spontaneous speech [Hain et al., 2006a]), which usually gives more performance improvement than enhancing the ASR subsystem [Abberley et al., 1998].

1.3.1 Approaches to STD

Designing an STD system needs to answer several questions: What kind of ASR system should be used? How to handle recognition errors? How to reject false alarms? Different answers to these questions lead to different systems that exhibit different properties. We summarise various approaches to a functional STD system in this section.

Word and subword -based systems

A natural and simple implementation of STD is a word-based system. In this implementation, the ASR subsystem is a LVCSR system that converts the input speech into word transcripts, or more often, word lattices. With the transcriptions or lattices, search terms can be detected using a simple search. Miller et al. [2007] used this approach in their STD system for the NIST 2006 evaluation, and achieved the best performance on CTS in English. Szöke et al. [2006] and Vergyri et al. [2007] described similar systems and all achieved good performance.

The word-based STD approach tends to achieve high accuracy because lexical information is utilised; however, it suffers a clear shortcoming in that out-of-vocabulary (OOV) terms can not be detected. We will define OOV terms thoroughly in Section 3.2.2; for now we roughly regard the words absent from the system dictionary as *OOV words*, and any terms containing OOV words as *OOV terms*. Correspondingly, words existing in the system dictionary are called *in-vocabulary (INV) words*, and terms containing only INV words are called *INV terms*. According to these definitions, OOV words never appear in the lattices generated by the word-based transcriber, and thus can not be discovered by the term detector. To address this problem, most researchers resort to systems based on subword units, usually phonemes. In a phoneme-based STD

system, the ASR sub-system is a phoneme recogniser that generates phoneme lattices, and the search terms are converted to phoneme sequences and are searched for in the lattices. The phoneme-based solution was first proposed for SDR, e.g., [Schäuble and Wechsler, 1995; Witbrock and Hauptmann, 1997; Wechsler et al., 1998; Ng, 2000; Cardillo et al., 2002; Ma and Li, 2005; Itoh et al., 2006; Ng, 1998], and has been applied to STD by many researchers, e.g., Szöke et al. [2006]; Wallace et al. [2007]; Parlak and Saraçlar [2008].

Besides phonemes, other subword units are also used in both SDR and STD, e.g., word-fragments [Seide et al., 2004], particles [Logan et al., 2005, 2002], acoustic words [Ma and Li, 2005], graphones [Akbacak et al., 2008; Vergyri et al., 2007], multigrams [Pinto et al., 2008; Szöke et al., 2008a], syllables [Meng et al., 2007], and graphemes [Wang et al., 2008a]. The basic idea of all these non-phoneme subword unites is to construct a suitable subword inventory that can balance the flexibility of representing novel words and the ability of capturing more lexical constraints.

Subword-based STD systems have been compared with word-based systems by a number of researchers, e.g., Szöke et al. [2005a]; Burget et al. [2006]; Šmídl and Psutka [2006]. Their conclusions are similar: word-based systems generally outperform subword-based systems when detecting INV terms, whereas subword-based systems can detect OOV terms that are usually important in real applications. Seide et al. [2004] reported that a phoneme-based system could reach the accuracy of a word-based system for INV terms, and the accuracy was nearly maintained for OOV terms, especially in the case that language models did not match the test domain.

To make use of the respective advantages of word and subword -based approaches, many researchers have developed combined systems that apply the word-based approach to detect INV terms, and the subword-based approach to detect OOV terms, e.g., James [1994, 1996]; Jones et al. [1996b]; Saraclar and Sproat [2004]; Parlak and Saraçlar [2008]; Iwata et al. [2008]; Szöke et al. [2006]. A hybrid approach was also proposed which fuses word and subword lattices, and then searches for both INV terms and OOV terms from the hybrid lattices [Yu and Seide, 2004; Meng et al., 2008b]. Some researchers took another hybrid approach, in which word/subword mixed lexica and LMs were built and applied to generates hybrid lattices [Yazgan and Saraclar, 2004; Akbacak et al., 2008; Szöke et al., 2008b].

Dealing with STD uncertainty

Uncertainty is an universal problem in speech research. For STD, the uncertainty mainly comes from two aspects: recognition errors and pronunciation variation. In the first aspect, recognition is far from accurate, which makes the transcription results unreliable; in the second aspect, people always pronounce words in different ways which means that the real pronunciation of words deviates from the lexical forms, unpredictably and stochastically. In spite of the underlying causes, the effect of these two kinds of uncertainty is the same: a deviation of the recognition results from the canonical transcripts, and thus can be addressed in the same way. Three approaches have been proposed to deal with the uncertainty: *lattice representation*, *query expansion* and *soft match*.

The lattice representation was introduced by James and Young [1994], and was adopted by many researchers, e.g., James [1994]; Brown et al. [1996]; James [1996]; Young et al. [1997]; Seide et al. [2004]; Yu and Seide [2004]; Thambiratnam and Sridharan [2005]; Akbacak et al. [2008]; Meng et al. [2007]. Lattices are a natural extension of 1-best transcriptions and n-best lists [Ng, 1998]. With lattices, alternative recognition hypotheses are retained along with various information such as time stamps, acoustic likelihood, language model scores, lattice density, context, etc. Lattices have been demonstrated working better than 1-best transcripts in keyword spotting and STD [Seide et al., 2004; Parlak and Saraçlar, 2008].

The second approach to address the uncertainty in STD is query expansion, which expands the original query by adding some terms that are similar or relevant. The 'similarity' or 'relevance' is measured by edit distance [Wechsler and Schäuble, 1995], acoustic confusion [Logan et al., 2005], model distance [Itoh et al., 2006] or semantic relevance [Hu et al., 2004].

The third method to compensate for the uncertainty is soft match, which allows a certain degree of mismatch in the term search. To penalise the mismatch, a cost is assigned to each detection, according to edit distance [James and Young, 1994; Thambiratnam and Sridharan, 2005; Itoh et al., 2006; Miller et al., 2007] or acoustic confusion [Wechsler et al., 1998; Srinivasan and Petkovic, 2000; Szöke et al., 2005a; Audhkhasi and Verma, 2007; Pinto et al., 2008] between the lexical form and the detected form of the search term.

Site	#Occurrences	Speech Hours	Occ. / Hours
BNEWS	4893	2.212	2211.66
CTS	5856	2.993	1956.78
CONFMTG	3672	2.098	1750.06

Table 1.1: The test conditions of the NIST 2006 STD evaluation. BNEWS stands for broadcast news, CTS is conversational telephone speech and CONFMTG is conference/meeting. This table is reproduced from the NIST 2006 STD evaluation report [Fiscus et al., 2006].

Site	Winner	ATWV	max-ATWV
BNEWS	IBM	0.8485	0.8532
CTS	BBN	0.8335	0.8336
CONFMTG	SRI	0.2553	0.2765

Table 1.2: The best results from the NIST 2006 STD evaluation. 'ATWV' is the average term-weighted value, and 'max-ATWV' is the optimal ATWV with an ideal decision approach; both are defined by NIST for STD evaluation [NIST, 2006]. Section 3.1.4 will discuss these metrics in detail. The numbers in the table are obtained from the NIST evaluation report [Fiscus et al., 2006].

Confidence estimation and decision making

Confidence estimation and decision making play an important role in STD for false alarm rejection. A widely used confidence measure is derived from lattice-based posterior probabilities [Wessel et al., 1998; Szöke et al., 2005a]. To accord with the NIST evaluation metric (ATWV), some researchers propose ATWV-oriented thresholds [Miller et al., 2007; Akbacak et al., 2008]. A neural network was used by Wallace et al. [2007] to map lattice-based confidence to classification posterior probability-based confidence, which is more suitable for false alarm rejection.

1.3.2 State-of-the-art results

The NIST 2006 STD evaluation provided official results of state-of-the-art STD systems [Fiscus et al., 2006]. We show the STD06 test conditions in Table 1.1 and the best results in Table 1.2. These results are the reference point for our experiments.

1.4 Challenges of out-of-vocabulary (OOV) words

STD faces many challenges, some of which are common to all ASR tasks, such as vulnerability to noise, channel variety and spontaneous speech phenomena, while some others are STD specific, such as real-time response, fast search, robust confidence estimation, recall & precision trade-off, etc. Among all these challenges, the issue of out-of-vocabulary (OOV) words is one of the most critical.

In the previous section, we mentioned that a subword unit-based approach can be used to solve this problem; however, the OOV issue is so crucial to STD that it deserves further discussion.

1.4.1 Origin of out-of-vocabulary words

Roughly speaking, OOV words are those words absent from the system vocabulary of a speech system. Some words are OOV because of the limited vocabulary size, but a large number of OOV words are newly coined by people everyday. In the book *Death Sentence, The Decay of Public Language*, Don Watson estimated that there are about 20,000 new words born each year, which is more than 50 per day [Watson, 2003]. These new words come from various sources in various domains, including (1) scientific and engineering terms, such as names of new medicine, new genes, new species, new stars, new methods, new concepts, etc; (2) new terms from social life, such as new trade marks, new products, new movie or sport stars, etc; (3) political terms, such as names of new government leaders, hot topics of debate or legislation; (4) foreign words borrowed from other languages. These new words constitute the major part of OOV words, and the number of them steadily increases as time goes on.

1.4.2 The OOV challenge in speech transcription

OOV words have been studied in speech transcription for a long time. Although they represent only a small fraction of the complete word list, they usually jeopardise system performance significantly. Because OOV words are not in the lexicon, OOV segments are always recognised as some INV words that are certainly incorrect; furthermore, the vicinity of the OOV segments tends to be misrecognised as well, leading to further recognition errors. Experimental results reported by Woodland et al. [2000] confirmed that ASR accuracy does relate to OOV rates directly.

To deal with OOV words in speech transcription, three levels of research has been

conducted by researchers: OOV detection, phoneme transcription and word recovery. We will discuss these research in turn.

OOV detection aims to detect OOV segments so that the adverse effect on neighbouring speech can be alleviated. Two approaches have been proposed: the first is a *filler approach* that constructs one or several filler models to absorb OOV segments [Sukkar and Wilpon, 1993; Weintraub, 1995]; the second is a *subword approach* that uses subword units to absorb OOV segments [Méliani and O'Shaughnessy, 1997; Bayya, 1998; Klakow et al., 1999; Bazzi and Glass, 2000b, 2001, 2002; Galescu, 2003; Yazgan and Saraclar, 2004; Bisani and Ney, 2005; Yamamoto et al., 2006].

Phoneme transcription takes a further step: it does not only detect OOV segments, but also transcribes these segments into phoneme strings. To conduct the recognition, both uniform language models [Bazzi and Glass, 2000a; Galescu, 2003; Bisani and Ney, 2005; Yazgan and Saraclar, 2004] and hierarchical decoding graphs [Bazzi and Glass, 2000b, 2001, 2002; Yamamoto et al., 2006] have been employed.

OOV word recovery is the most difficult task, which not only transcribes the OOV segments, but also tries to guess the spellings of the unknown words. Two approaches have been investigated: in the two-stage approach, OOV segments are first converted to phoneme strings and then are transformed to spellings [Decadt et al., 2001]; in the one-stage approach, spellings and phonemes are obtained together with a joint-multigram model [Galescu and Allen, 2001].

1.4.3 The OOV challenge in STD

In STD, we define a term containing one or more OOV words as an *OOV term*. OOV terms present a big challenge to STD, even more serious than to transcription, because STD is an open vocabulary task, meaning that users may issue queries containing any words. Logan et al. [2000] reported that in a real spoken document retrieval system, 12% of queries contained OOV terms. These OOV terms often convey the central request of the query, and therefore must be seriously addressed. Furthermore, with new words devised every day, the OOV problem will become more and more serious as time goes on, if it is not particularly addressed. We can imagine that after one or two years, unpopular words become popular, newly coined words are widely used, social events come out one after another... If the system does not consider the OOV problem in its design, it will become out-of-date quickly, no matter how big a lexicon is used.

Basically, the challenges that an STD system faces when detecting OOV terms

arise from three aspects: uncertainty in pronunciation, high diversity in properties, and weakness in modelling.

Pronunciation uncertainty Pronunciation uncertainty is more serious for OOV terms. Firstly, the pronunciations of OOV terms are unknown, and thus must be predicted by some letter-to-sound (LTS) approaches. Unfortunately, all LTS approaches reported so far suffer from a high word error rate of typically about 30-40% or phoneme error rate of about 10% [Torkkola, 1993; Deligne et al., 1995; Luk and Damper, 1996; Damper and Eastmond, 1997; Black et al., 1998; Daelemans et al., 1999; Bisani and Ney, 2003b; Taylor, 2005]. The error-prone pronunciation prediction makes the OOV detection based on unreliable lexical forms. Furthermore, the pronunciations of OOV terms exhibit more variation than those of INV terms, because people need to guess the pronunciation when encountering an unfamiliar word. In this process, we tend to slow down, examine the spelling structure, guess the pronunciation, make hesitations, and then try to pronounce. This guess-and-trial process leads to more spontaneous speech phenomena and more acoustic variation. More seriously, guessing pronunciations is not always easy; for unusual terms, different people might guess differently, leading to pronunciation variation at the lexical level. The interweaving of variations at the acoustic level and the lexical level makes OOVs more difficult to deal with.

Property diversity Different OOV terms possess very different properties, e.g., occurrence rate, phonemic structure, linguistic background, morphological form, etc. This diversity makes it difficult to design a detection scheme that is suitable for all types of terms.

Weak modelling OOV terms tend to be weakly modelled by acoustic models (AMs) and language models (LMs) since they have no instance in the training data. The consequence is that lattices tend to miss OOV terms, and confidence measures derived from AM and LM scores are unreliable.

1.5 Motivations and hypotheses

Considering its fundamental importance, research on the OOV issue is till limited. The subword-based approach that is commonly used to deal with OOV terms actually treats OOV terms no differently to INV terms except that the pronunciations are obtained by

different ways. The special properties of OOV terms have never been seriously treated. This is obviously not ideal.

In this thesis, we study OOV term detection, paying particular attention to the special properties of OOV terms. The special properties that we consider correspond to the three OOV challenges discussed in the previous section, i.e., pronunciation uncertainty, property diversity and weak modelling. To the author's best knowledge, there has been no STD research directly addressing these special properties thus far.

Specifically, we make and test three hypotheses in this thesis:

- A stochastic pronunciation model based on joint-multigrams can compensate for the uncertainty in pronunciations of OOV terms;
- A term-dependent hit/FA decision strategy based on discriminative models can cope with the high diversity of OOV terms;
- A discriminative model-based confidence estimation can alleviate the weak modelling of OOV terms.

We hope to answer four questions: 1. *How to predict pronunciations of OOV terms;* 2. *How to handle pronunciation variation of OOV terms;* 3. *How to treat the high diversity among OOV terms;* 4. *How to estimate confidence that is robust for OOV terms.* Answers to these questions will solve the OOV-induced challenges that we discussed in the previous section.

1.6 Contributions and publications

This thesis makes three contributions to STD, particularly for OOV term detection. Firstly we propose a stochastic pronunciation model (SPM) based on joint multigrams to compensate for OOV pronunciation uncertainty by considering multiple pronunciations; secondly we propose a term-dependent decision strategy based on discriminative models to deal with OOV diversity by integrating term-dependent decision factors; thirdly we propose a direct posterior confidence estimation that replaces generative models such as hidden Markov models (HMMs) with discriminative models such as multiple layer perceptrons (MLPs), so that weak modelling is alleviated.

A number of publications have resulted during the study for this thesis, as listed in the following.

1. [Wang et al., 2008a] We studied phoneme and grapheme -based STD and proposed detection combination.

2. [Wang et al., 2009a] The stochastic pronunciation model was proposed in this paper.
3. [Wang et al., 2009b] The term-dependent discriminative decision strategy was proposed in this paper.
4. [Wang et al., 2009c] The direct posterior confidence measurement was proposed in this paper.

More publications result from exploring studies and collaborated work. These studies are beyond the scope of this thesis, and therefore are not included here.

1. [Tejedor et al., 2008] In this paper, we studied grapheme-based STD in Spanish. This is a collaborative work with Javi Tejedor.
2. [Frankel et al., 2008] This is a collaborative work with Joe, in which we studied a posterior probability feature based on bottleneck MLPs.
3. [Wang et al., 2008b] This is a collaborative work with Ivan Himawan, in which we proposed a posterior beamforming for microphone array-based ASR.
4. [Tejedor et al., 2009] We proposed in this paper to combine phoneme and grahpeme -based systems based on direct posterior confidence estimation. This is a collaborative work with Javi Tejedor.

1.7 Organisation of the thesis

The whole thesis will focus on designing novel techniques for OOV term detection. Chapter 2 reviews related research and Chapter 3 describes the experimental settings. Then we present our implementation of the joint-multigram model in Chapter 4 and the stochastic pronunciation modelling based on this model in Chapter 5. Following that, Chapter 6 proposes the discriminative decision strategy and Chapter 7 proposes the direct posterior-based confidence estimation. Finally the work of this thesis is summarised in Chapter 8, along with some ideas for future research.

Chapter 2

Literature review

In this chapter, we review in more depth previous studies related to the topic of this thesis. We first review research on keyword spotting and spoken term detection, and then focus on the subjects of this thesis, including pronunciation prediction, uncertainty treatment and confidence estimation.

2.1 Keyword spotting

STD research closely relates to keyword spotting, so we start our review 36 years ago when Bridle presented the first keyword spotting system [Bridle, 1973].

2.1.1 From templates to hidden Markov models

Template-based keyword spotting

Bridle's system [Bridle, 1973] searched keywords by matching the keywords' templates to the test speech. By sliding the starting frame of the match, potential keyword occurrences were detected using the famous dynamic time warping (DTW) algorithm. This template-based approach was adopted by many researchers in the 1980s, including Christiansen and Rushforth [1977]; Myers et al. [1980]; Higgins and Wohlford [1985].

HMM-based keyword spotting

An obvious weakness of the template-based approach is that templates can not represent the inherit variation of human speech. This disadvantage arises from the approach

itself and can not be amended even with multiple templates [Christiansen and Rushforth, 1977]. To overcome this problem, Wilpon et al. [1989] from AT&T and Rohlicek et al. [1989a] from BBN independently and simultaneously presented a novel approach based on hidden Markov models (HMM) in 1989. In implementation, Wilpon et al. [1989] used the frame-sliding approach to detect potential keywords while Rohlicek et al. [1989a] used alternative models to 'absorb' non-keyword speech. In spite of this difference, they laid a foundation for the HMM-based keyword spotting together.

Architecture of HMM-based keyword spotting

To improve the HMM-based keyword spotting, Wilpon et al. [1990] introduced garbage models to represent extraneous speech and background noise. The garbage model corresponds to the filler model in [Rose and Paul, 1990]. Various filler models have been proposed [Rose and Paul, 1990; Manos and Zue, 1997], and the best performance was reported with triphones [Rose and Paul, 1990]. An on-line garbage model was proposed in [Boite et al., 1993; Bourlard et al., 1994] which took the average score of n-best paths in decoding as the score of garbage.

Meanwhile, Rose and Paul [1990] proposed a background model to normalise the acoustic scores of detected keywords, which makes the scores of detections from different utterances comparable. Background models were further studied in [Lleida et al., 1993; Jeanrenaud et al., 1993]. Junkawitsch et al. [1996] normalised the path scores in decoding, which is equivalent to an implicit background model applied online.

Rahim et al. [1995] introduced anti-word models, which were trained for each keyword, with training data comprising all other keywords. Generally speaking, background models and anti-word models both aim to normalise incomparable confidence scores, though the background model works by smoothing while the anti-word model works by contrasting.

This research established the standard architecture for a keyword spotting system, as shown in Figure 2.1, where the filler network represents extraneous speech and the background network normalises the confidence score. By this architecture, the filler network is more related to modelling while the background network is for confidence estimation. For example, in a phoneme-based spotting system [Szöke et al., 2005b], the filler network is a phone loop, while in a LVCSR-based spotting system [Weintraub, 1995], the filler network represents noise and short pause only. Similarly, with a likelihood ratio-based confidence [Rose et al., 1995], the background network com-

prises anti-word models, while with a lattice-based posterior confidence [James and Young, 1994], the background network is just the keyword network.

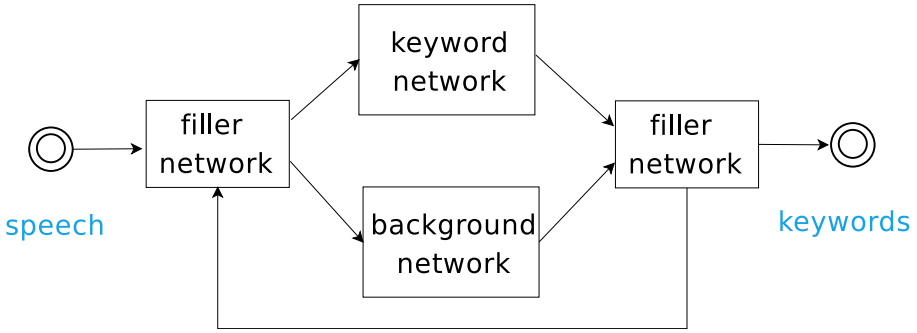

Figure 2.1: The diagram of a standard keyword spotting system. The two double circles represent the starting and ending states respectively.

In early spotting systems, keywords were usually represented by word models, e.g., [Wilpon et al., 1989; Rohlicek et al., 1989a; Rose and Paul, 1990; Lleida et al., 1993]. Word-based modelling tends to give high detection performance, however it makes adding new keywords difficult. To solve this problem, most recent systems chose subword models to represent keywords, e.g., [Foote et al., 1995; Méliani and O'Shaughnessy, 1997; Szöke et al., 2005b]. Similarly, filler models and background models were usually based on words [Rose and Paul, 1990; Lleida et al., 1993; Boite et al., 1993] or lexical words [Jeanrenaud et al., 1993] in the early days, but are more often based on subword units today, such as context-independent phones [Rose and Paul, 1990; Boite et al., 1993; Jeanrenaud et al., 1993], context-dependent phones [Rose and Paul, 1990; Lleida et al., 1993; Manos and Zue, 1997; Szöke et al., 2005b], syllables [Lleida et al., 1993; Méliani and O'Shaughnessy, 1997] or phone classes [Rose and Paul, 1990; Boite et al., 1993].

Landmark-based keyword spotting

Recently, Jansen and Niyogi [2008] proposed a landmark-based system which employs point process models on acoustic events. The authors reported that even with noisy and extremely sparse training data, the phone landmark-based approach achieved an accuracy level comparable to the HMM-based approach.

2.1.2 Two-stage keyword spotting

Figure 2.1 shows an *integrative* approach which detects keywords as the speech is processed. This approach is efficient for applications such as keyword monitoring or command control. In other applications that have to search a huge amount of speech and provide a response in a short time, the integrative approach becomes too slow to be acceptable. To solve this problem, we can transcribe the speech into some intermediate representation offline, and then search for queries within the intermediate representation. Because the costly speech recognition has been conducted beforehand, the on-line query can be very fast. This leads to a *two-stage* approach, as illustrated in Figure 2.2.

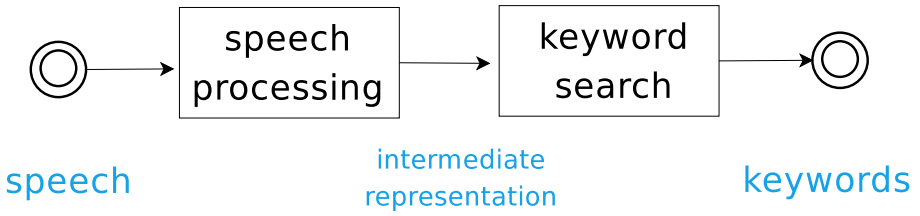

Figure 2.2: The two-stage keyword spotting approach in which speech data are first converted to intermediate representation and then keywords are searched for in the intermediate representation.

1-best approach

Some researchers use 1-best LVCSR transcripts as the intermediate representation [McDonough et al., 1994; Wactlar et al., 1996; Dharanipragada and Roukos, 1997; Witbrock and Hauptmann, 1997; Abberley et al., 1998; Logan et al., 2000; manuel Van Thong et al., 2000; Thong et al., 2002], while others use phoneme transcripts either generated by phoneme recognition [Schäuble and Wechsler, 1995; Wechsler and Schäuble, 1995; Hauptmann et al., 1998; Wechsler et al., 1998; Ng, 1998] or converted from word transcripts [Witbrock and Hauptmann, 1997; Amir et al., 2001; Logan et al., 2005]. Besides phonemes, other subword units are also used in the 1-best approach, such as word-fragments [Ramabhadran et al., 2009; Seide et al., 2004], particles [Logan et al., 2002, 2005], acoustic words [Ma and Li, 2005], graphemes [Billa et al., 2002] or sub-phonetic segments (SPS) [Itoh et al., 2006].

N-best and lattice-based approaches

Instead of using 1-best transcripts, Ng [1998]; Hauptmann et al. [1998] used n-best transcripts as the intermediate representation. James and Young [1994] extended the n-best approach to a lattice-based approach, in which he used phone lattices as the intermediate representation, and detected the desired keywords from lattices using symmetric dynamic programming [Young et al., 1997; Thambiratnam and Sridharan, 2005; Thambiratmann and Sridharan, 2007]. Mismatch was allowed, to compensate for recognition errors, and confidence was measured as the likelihood ratio of the path corresponding to the detected keyword to the best path in the time interval of the detection.

Compared to 1-best and n-best transcripts, lattices retain more information of the original speech; moreover, lattices can be represented in a compact way and efficient search algorithms are available. For these reasons, the lattice-based approach has been adopted by many researchers in keyword spotting, STD and SDR. For example, Brown et al. [1996]; Young et al. [1997] used phoneme lattices in voice mail retrieval (VMR); Saraclar and Sproat [2004] used word lattices and phoneme lattices for SDR. Seide et al. [2004] compared word, phoneme and word-fragment lattices; Szöke et al. [2005a] compared an integrative spotting system and two word and phoneme lattice-based spotting systems.

A special type of lattice that contains no overlapped arcs, called *confusion network* [Mangu et al., 1999], was used by Woodland [2000] in keyword spotting and showed to be superior to regular lattices. Another variant of lattice, the *position specific posterior lattice* (PSPL) that retains posterior probabilities on the arcs, was proposed by Chelba and Acero [2005] for efficient indexing and ranking for STD and SDR.

Hybrid approach

Some of the above approaches can be combined. For example, Jones et al. [1996a] combined the LVCSR 1-best approach and the integrative keyword spotting for VMR, where the keyword spotting handled search terms not in the lexicon of the LVCSR system. Combining the LVCSR 1-best approach and the phoneme lattice-based approach was proposed by James [1996] for broadcast news retrieval, and Jones et al. [1996b] for VMR.

Yu and Seide [2004] combined word and phoneme lattice-based systems either by a *prior combination* based on a word-subword hybrid recognition, or by a *posterior*

combination that merged detections from word and phoneme-based systems. Their experimental results showed that INV words and OOV words behaved differently with these two combinations, and it is hard to tell which method is better.

The prior combination was further studied in [Akbacak and Hansen, 2006b; Akbacak et al., 2008; Ramabhadran et al., 2009], and the posterior combination was further studied in [Mamou et al., 2008; Iwata et al., 2008]. Akbacak and Hansen [2006a] compared the prior and posterior combination as well, but what they combined was two kinds of lattices that were generated with two phone sets from different languages. The authors reported consistent advantage with the prior combination.

2.1.3 Fast search

To speed up the search of queries, inverted indexes are usually constructed from transcripts or lattices, which was theoretically described by Allauzen et al. [2004]. In implementation, Cardillo et al. [2002] proposed an index structure called a *phonetic track*, and Burget et al. [2006] used an index based on phone trigrams. Siohan and Bacchiani [2005] proposed path-based indexes based on word-fragments derived from LMs with pruning applied. Yu et al. [2005] proposed to use M-gram phoneme LMs to approximate the expected term frequencies (ETF) of query terms so that the overall data-access efficiency was optimised.

A more pragmatic strategy is the *zoom-in* approach. For example, Yu and Seide [2005] proposed a two-stage detection that first retrieves speech segments containing search terms from the inverted index, and then performs linear search within the lattices of the spotted segments; a similar approach was proposed in [Dharanipragada and Roukos, 2002; Itoh et al., 2006], except that the exact match is performed using integrative keyword spotting.

2.1.4 Discriminative training and modelling

Discriminative training for keyword spotting was first proposed by Rose [1992], and is similar to training with respect to a maximum mutual information (MMI) criterion, implemented as a gradient descent procedure. This idea was extended by Sukkar and Wilpon [1993] into a two-pass discriminative processing: the first pass minimises the classification errors with a generalised probabilistic descent (GPD) discriminative training, and the second pass applies Fisher's discriminant analysis. The following research investigated various discriminative training techniques and models, e.g., min-

imum classification error (MCE) oriented training with GPD [Rahim et al., 1995], linear discriminative functions [Lleida et al., 1993], neural networks [Lleida et al., 1993; Fernández et al., 2007; Wöllmer et al., 2009], large margin approaches [Grangier, 2008; Keshet et al., 2009]. The discriminative approach was recently applied to train the decoding graph of a finite state transducer (FST) -based speech indexing system [Chaudhari et al., 2008].

2.1.5 Query by example

Normally, a keyword spotting system searches for a keyword from its written form; in some circumstances, however, keywords have to be searched by voice, e.g., in a telephone-based information retrieval system, it is more convenient to issue a query by speaking. This can be done with template matching using the DTW algorithm [Bridle, 1973], HMM-based enrolment [Wilcox and Bush, 1992], or baseform generation [Ramabhadran et al., 1998]. Very recently, lattice alignment was presented by Lin et al. [2008, 2009] which matches the lattices of the query speech and the searched speech. Shen et al. [2009] proposed a novel approach based on posteriorgram match which is reminiscent of the DTW approach, except that the speech is represented by posteriorgrams.

2.2 Spoken term detection

In fact, STD is just another name for two-stage keyword spotting, except that new metrics are applied in evaluation, particularly the Actual Term-Weighted Value (ATWV). This section will review research that follows the NIST 2006 STD evaluation standard. Some research that does not strictly follow the NIST standard (e.g., those did not report their results in terms of ATWV), is also included if it essentially belongs to the two-stage term search.

Because the basic technologies of STD have already been summarised in the discussion of two-stage keyword spotting, we will now introduce contemporary work on STD organised by research institute.

Brno University of Technology (BUT)

BUT's research on STD is based on their work on acoustic keyword spotting [Szöke et al., 2005b], lattice-based term search [Szöke et al., 2005a] and document retrieval

[Burget et al., 2006]. In the NIST STD 2006 evaluation, BUT's system achieved a good ranking in all three conditions [Fiscus et al., 2006; Szöke et al., 2008c]. In this system, word lattices generated with 4-gram word LMs and phoneme lattices generated with 2-gram phoneme LMs, after posterior pruning, were used for INV and OOV detection respectively. Inverted indexes were constructed with ungirams for the word system and 3-grams for the phoneme system. Lattice-based posterior probabilities were used as the confidence measure, with normalisation by the duration of the detection and the number of phonemes of the query. They suggested that a large branching factor in decoding plus strict lattice pruning tends to improve detection quality.

Word and subword models were studied in [Szöke et al., 2008a], and word-subword hybrid systems were studied in [Szöke et al., 2008b]. In these studies, they experimented with various subword units, and reported that multigrams developed on a LVCSR vocabulary gave the best performance. Moreover, they reported better performance with the hybrid system, for both INV terms and OOV terms. Different from the results in [Akbacak and Hansen, 2006a], their hybrid system achieved worse accuracy than the posterior combination, but generated more compact lattices and accomplished faster detection.

IBM research

In the NIST 2006 evaluation, IBM achieved the first position on BNEWS, and the second on CTS and CONFMTG. Their system utilised word confusion networks for INV term detection and 1-best phoneme transcripts for OOV term detection [Mamou et al., 2007]. For INV terms, the confidence score was calculated as the product of the posterior probability of the detected term and a scale factor according to its rank in the confusion list. For OOV terms, the confidence was derived from the time gaps between phonemes of the detection.

Ramabhadran et al. [2009] extended this work with a fuzzy match on word-fragment transcripts. A neural network-based confusion matrix was used for the fuzzy search, and a word-word fragment mixed lexicon was used to develop a hybrid system. Results showed that the hybrid system was faster than combing individual word and phoneme -based systems, but the performance was significantly degraded when the ASR accuracy decreased for fast decoding and detection.

The STD system was further migrated to SDR [Mamou et al., 2008], with some new extensions including word-fragment indexes and fuzzy search. Significantly, they provided a phonetic query expansion which utilises the n-best pronunciations predicted

by the joint maximum entropy N-gram model [Chen, 2003]. This technique was further discussed by the same authors in [Mamou and Ramabhadran, 2008], where confidence scores of the n-best predicted pronunciations were based on posterior probabilities. In another joint work [Can et al., 2009], the phonetic expansion was applied to STD based on weighted finite state transducers (WFST). The authors indicated that proper weights are important for the query expansion. This method is similar to the stochastic pronunciation modelling we propose in Chapter 5, however we used a joint-multigram model which allows variable-sized grapheme-phoneme correspondence.

Microsoft Research Asia

Microsoft Research Asia is another major group working on STD research. They largely extended the lattice-based approach [Yu and Seide, 2004] and fast search [Yu et al., 2005], and enhanced STD with standard text indexers [Seide et al., 2008].

Their recent study focused on Chinese STD. In [Meng et al., 2007], they compared lattices based on various units: word, character, tonal syllable and toneless syllable, and studied various methods for lattice conversion. The experimental results showed that toneless syllable lattices converted from word lattices resulted in the best performance. They also explored lattice amendment with extra arcs, as well as system combination.

To handle OOV queries, they proposed an interesting two-stage approach [Meng et al., 2008a]. In this approach, word lattices are first converted to toneless syllable lattices so that OOV terms can be searched for by fuzzy matching. After that, the syllables of the OOV term are inserted into the syllable lattices at the detected positions, with the acoustic and LM scores calculated from the acoustic model and language model. Finally, the 'amended' lattices are re-scored to get the confidence of the detected OOV terms. A performance gain was reported with this approach, though an obvious shortcoming is that the original speech has to be re-accessed.

Lattice-based combination was further studied by the same authors in [Meng et al., 2008b]. In this work, they introduced a time-based lattice compression approach which merges lattice nodes that are close in time. This approach reduced the lattice size dramatically, but did not jeopardise STD performance, partly due to the additional arcs between adjacent nodes.

BBN technology

BBN has a long-standing reputation in keyword spotting. They originally proposed the HMM-based spotting approach [Rohlicek et al., 1989b], and contributed many novel techniques such as discriminative training [hung Siu et al., 1997], segmental-based feature extraction [Gish et al., 1992] and confidence estimation [Jeanrenaud et al., 1995; Siu and Gish, 1999].

In the NIST 2006 evaluation, BBN's system achieved the best performance on CTS in English [Fiscus et al., 2007]. In this system, they used word lattices for INV term detection and phoneme transcripts derived from 1-best word transcripts for OOV term detection. Word lattices were indexed for fast match. Each word had a list of detection candidates sorted in order of confidence derived from lattice-based posterior probabilities. OOV terms were searched for in the phoneme transcripts, with confidence based on edit distance.

A novel feature of their system is an ATWV-oriented decision strategy. In this strategy, the confidence threshold is selected such that the hit/FA trade-off is optimised with respect to ATWV. Note that in their STD06 system, this technique was only applied to detect INV terms; for OOV terms, some simple assertion rules were designed and examined in hit/FA decision. In Chapter 5 we propose a confidence normalisation technique which was motivated by the ATWV-oriented decision. The advantage of our technique is that the normalisation is based on discriminative confidence measures, thus avoiding possible bias [Siu and Gish, 1999].

SRI

The SRI system [Vergyri et al., 2006] achieved a performance in the leading group of the NIST 2006 evaluation. This system was based on word lattices and N-gram indexing. Word 1-grams to 5-grams were dumped with associated information, subject to posterior-based pruning. A novel feature of their system is an ANN-based confidence mapping which leads to a discriminative decision. This is similar to our work in Chapter 6, though we focus on term-dependent decision factors and investigate various discriminative models. Moreover, the discriminative decision approach is combined with confidence normalisation in our work, leading to an ATWV-oriented decision making.

A major shortcoming of their system is that OOV terms were not covered. To handle OOV terms, they proposed a graphone-based hybrid system [Akbacak et al., 2008]. In this system, graphones were generated by a joint-multigram model trained

on a large dictionary. The frequent graphones were appended to the word lexicon, and the language model was re-trained with the text corpus in which OOV terms were converted to graphone sequences by the joint-multigram model. The hybrid lexicon and LM were used to perform decoding and generate hybrid lattices, from which both INV and OOV terms were detected.

OGI

OGI provided another leading STD system in the NIST 2006 evaluation. Different from many other systems, their system was based on finite state transducers (FSTs) [Shafran et al., 2006; Vergyri et al., 2007]. In their implementation, the word lattices provided by SRI were converted to n-gram finite state machines (FSMs) which were then converted to FSTs. A transducer index was created by the union of all the utterance-level transducers. In term search, if the query was INV, it was first converted to a FSM, which was then matched to the transducer index to retrieve the potential occurrences; on the other hand, if the query was OOV, it was first converted to a pronunciation and then was matched by a phone-to-word transducer that was learnt from the ASR lattices. A SVM was applied for re-scoring. The ATWV-oriented decision strategy was also proposed [Vergyri et al., 2007], similar to BBN [Fiscus et al., 2007].

Queensland University of Technology (QUT)

Although QUT achieved a relatively low performance in the NIST 2006 evaluation, it is still a leading group in STD. The low performance, to our mind, might be caused by the pure phoneme-based approach and the term-independent decision strategy.

QUT's STD06 system [Wallace et al., 2007] was purely based on phoneme lattices. A Viterbi traversal was applied to convert phoneme lattices to phoneme n-grams in indexing. A dynamic match lattice spotting (DMLS) algorithm [Thambiratmann and Sridharan, 2007] was proposed to perform the lattice traverse, based on on minimum edit distance (MED). A neural network was proposed to fuse the MED score and other term-dependent features to give a composite confidence. This approach is similar to the discriminative decision strategy we will propose in Chapter 6, though our work is based on a more general framework that allows any discriminative models and is compatible with the evaluation metric, ATWV.

Their recent work [Wallace et al., 2009] focuses on rapid detection. To speed up the term search, they used a monophone-based ASR system to accelerate speech tran-

scribing, and a two-tier hierarchical search to accelerate term detection.

Other institutes

Besides the studies described above, many other research groups have presented novel work recently. Pinto et al. [2008] described a very fast STD system based on phoneme sequences, allowing confusion matrix-based mismatch; Dubois and Charlet [2008] applied a phoneme sequence search as well, but obtained phoneme transcripts from LVCSR transcripts. Parlak and Saraçlar [2008] studied STD on Turkish broadcast news, and reported a word-morph combined system; Mertens and Schneider [2009] studied STD on German, and compared 1-best search and lattice search based on syllables.

John Hopkins University (JHU) is actively working on multilingual STD [Sproat et al., 2008]. They described an STD system that can detect foreign terms using pronunciations derived from LTS, generated according to phone confusions, or even retrieved from the Internet. Their system, based on FSTs, achieved better performance on foreign terms. Other research focusing on the multilingual phenomenon includes multilingual SDR [Akbacak and Hansen, 2006b] and multilingual name entity retrieval [Akbacak and Hansen, 2006a].

2.3 Pronunciation prediction

Predicting pronunciations for novel words has been studied for more than 30 years, in the name of letter-to-sound (LTS) conversion or grapheme-to-phoneme (G2P) conversion, mostly to meet the request of TTS systems for unknown word synthesis. All the research can categorised as either rule-based or data-driven.

2.3.1 Rule-based approach

The rule-based approach predicts pronunciations by applying a set of *manually* designed linguistic rules. Re-write rules [Chomsky and Halle, 1968] and two-level rules [Kaplan and Kay, 1994] have been widely used in early LTS systems, e.g., DECTalk [Hallahan, 1995] and MITalk [Quinlan, 1992]. Klatt [1987] gave a good review on issues of designing a rule-based system, and compared some early rule-based systems. More recently, Divay and Vitale [1997] proposed a refined rule-based LTS system for

English and French; Kim et al. [1998] presented a similar system for Korean considering morphonemes.

Meng and colleges applied a hierarchical parsing tree to integrate knowledge sources at various levels (e.g., phonetic, phonemic, phonotactic, syllabic and morphemic level) and constructed statistic models at each level. Meng's method [Meng et al., 1994, 1996; Meng, 2001] is actually a hybrid approach that combines linguistic knowledge and statistical learning based on a tree structure.

2.3.2 Data-driven approach

Although the rule-based approach can achieve a fair performance, it suffers some inherent shortcomings: first, designing rules by hand is costly; second, rules must be applied in a specific order to avoid conflicts, which makes it difficult to add new rules; third, rules describe general phonological and phonotactic principles, thus are not suitable for predicting words like proper names and foreign words.

To address these problems, data-driven approaches were proposed. These approaches can learn both phonological rules and special cases from training data, and usually provide higher prediction accuracy. According to the manners in which the training data are used, the data-driven approaches can be classified as memory-based approaches or model-based approaches.

2.3.2.1 Memory-based approaches

The memory-based approach remembers all the training exemplars and predicts pronunciations for novel words by recalling the training exemplars in memory, and therefore is also called *lazy learning*. The underlying rationale is the belief that "intelligent performance is the result of the use of memories of earlier experience rather than the application of explicit but inaccessible rules" [Stanfill and Waltz, 1986].

The memory-based approach can be implemented as instance-based learning or analogy-based learning, according to how training exemplars are learnt.

Instance-based learning Instance-based learning [Aha et al., 1991] is also referred as *case-based learning*, *exemplar-based learning*, *similarity-based learning*, etc. In this learning method, grapheme and phoneme sequences of each entry in the training dictionary are aligned first, and then each grapheme-phoneme pair plus its grapheme context constitutes a *training exemplar*. Afterwords, training exemplars are organised

into some compact structure that can be searched efficiently. Finally, pronunciations of a novel word can be predicted by searching for the pronunciation of each grapheme of the word and concatenate them together.

Training exemplars can be structured in different ways in memory. It could be either a flat memory as in instance-based learning with an information grain criterion (IB1-IG) [Bosch and Daelemans, 1993] or a context table in the table-lookup method [Bosch and Daelemans, 1993]. A more poplar structure is a decision tree, which is compact in memory and improves searching efficiency. For example, Torkkola [1993] used a tree structure to save dynamically expanding contexts (DEC), Daelemans and Bosch [1993] used decision trees in their TABTalk system, Andersen and Dalsgaard [1994] used the same approach in Trie search.

The importance of a context grapheme various according to its position, and thus should be assigned different weights. DECTalk and other early systems assigned these weights by hand; more recent systems use criteria derived from information theory. For example, Daelemans et al. [1997] used information gain (IG) to determine the feature weights in IB1-IG. Information gain was also used by the same authors to determine the order in which contexts are examined in IGTree.

During prediction, if a test case was seen in training, its pronunciation can be retrieved directly from the exemplars; otherwise, the pronunciation is produced either from non-terminal nodes of decision trees [Torkkola, 1993; Daelemans et al., 1997] or content of default tables [Bosch and Daelemans, 1993], or determined by a majority decision of similar exemplars [Daelemans et al., 1998].

Note that trees here are used to compress memory without any abstraction. Any pruning will cause information loss, leading to abstract models that are no longer memory-based.

Analogy-based learning Instead of predicting pronunciations of individual graphemes, we can also predict pronunciations of grapheme strings, or *word fragments*. This idea leads to analogy-based learning.

The rationale of analogy-based learning lies in the intuitive assumption that we tend to guess the pronunciation of a new word by assembling the pronunciations of word fragments that we are familiar with. Psychological evidence for this was published by Glushko [1979], and the first analogy-based LTS system was presented by Dedina and Nusbaum [1986] in their PRONOUNCE system. In that system, fragments of the word under prediction were found by searching amongst those *memorised* exemplars, based

on which a pronunciation lattice was constructed by assembling the pronunciations of the found fragments. Finally, the best pronunciation was obtained by finding the best path through the pronunciation lattice. Damper and colleagues followed this line and extended Dedina's algorithm with multiple heuristic scoring [Sullivan and Damper, 1990; Damper and Eastmond, 1997; Marchand and Damper, 2000]. More recent work reported by Bellegarda [2005] considered orthographic neighbours in analogy learning with latent semantic analysis (LSA).

Comparing instance-based learning with analogy-based learning, the latter approach was reported to give better performance [Damper et al., 1998], possibly because it learns more information from word segments than single graphemes.

2.3.2.2 Model-based approaches

Instead of remembering all the training exemplars as in the memory-based approach, the model-based approach constructs abstract models by learning from the training exemplars. These models, if well trained, hopefully represent the language's phonological principles that can be applied to predict pronunciations for novel words. Among various models, computational rules, neural networks, decision tress, finite state transducers and joint-multigram models are commonly used.

Computational rules Context-dependent rules can be extracted from training exemplars in the form $\{g_i -> q_i|C_i\}$, where g_i and q_i are the grapheme under prediction and its corresponding phoneme respectively, and C_i denotes the grapheme context of g_i. These computational rules have the same form as those in the rule-based approach, except that they are learnt from data instead of being designed by a human. A simple and straightforward implementation was reported by Klatt and Shipman [1982], and an extended version was described by Bagshaw [1998], where dynamic contexts were considered. Hochberg et al. [1991] reported similar work but organised the induced rules in a hierarchical way.

Artificial neural network The first application of artificial neural networks (ANN) to LTS was proposed by Sejnowski and Rosenberg [1987] in the NETtalk system. In their work, the grapheme under consideration and its 6 neighbours were fed into a multiple-layer perceptron (MLP), which output a set of articulatory features that were mapped to a phoneme. This approach was reimplemented by McCulloch et al. [1987] in the Netspeak system and was refined by Jensen and Riis [2000] with a better coding

scheme. Adamson and Damper [1996] used a different type of ANN – a recurrent network, so that non-aligned data could be handled.

Decision tree We mentioned in the instance-based approach that trees can be used to organise training data efficiently. In the model-based approach, a tree is used for more than saving memory, but for representing a decision process, and so is an implicit representation of the phonological knowledge learnt from the data.

In that sense, the DEC tree [Torkkola, 1993] and Trie tree [Andersen and Dalsgaard, 1994] can be regarded as the simplest decision trees that split the intermediate nodes according to the grapheme contexts. IGTrees are a little more complex: they determine the order of the context application according to information gain [Daelemans et al., 1997]. A more complex tree is built with the ID3 and its successor the C4.5 algorithm [Quinlan, 1992], which determines the next questioned context in a dynamic way as the tree grows. This approach was adopted by many researchers, e.g., [Pagel et al., 1998; Suontausta and Häkkinen, 2000; Häkkinen et al., 2003]. Another type of tree is the classification and regression tree (CART), proposed by Lucassen and Mercer [1984] and thoroughly studied by Breiman et al. [1984]. This kind of tree is grown by testing a set of binary questions for each node, and choosing the best question, in terms of entropy decrease, to split the current node into two child nodes. These questions can be either manually designed or automatically learnt from data [Lucassen and Mercer, 1984]. Jiang et al. [1997] tried to improve the prediction accuracy of a CART by re-scoring n-best predictions with phonemic trigrams. Pagel et al. [1998] presented that considering part of speech tags might be helpful. Black et al. [1998] implemented the CART-based LTS approach in the Festival text-to-speech (TTS) system. Kienappel and Kneser [2001] proposed to compact trees using group questions that were learnt from data.

Hidden Markov model (HMM) HMMs were first used to perform LTS conversion by Parfitt and Sharman [1991]. In their implementation, phonemes were regarded as latent variables that obey the first-order Markov assumption, and graphemes were treated as observations that were conditionally independent. The task of predicting the pronunciation of a word amounts to finding the optimal state sequence in the HMM, given the spelling of the word as the observation. Although this model is theoretically feasible, the first-order assumption among phonemes and the conditional independence assumption among graphemes are incorrect, and therefore unsurprisingly this method

gives worse performance than other approaches such as the decision tree. Recently, Taylor [2005] reimplemented this model and applied some thoroughly designed pre-processing to improve the prediction accuracy.

Finite state transducer It has been well known that both context-dependent rules and two-level rules define a regular relation that equals a finite state transducer (FST) under certain assumptions and so rule-based prediction can be implemented as a FST match [Kaplan and Kay, 1994]. To represent statistical models, FSTs can be extended to stochastic phonographic transducers (SPTs) [Luk and Damper, 1996]. In general, manually designed rules, computational rules, decision trees, and HMM can all be written as FSTs. Using the FST as a universal representation, different prediction approaches relying on different knowledge sources can be integrated into a single framework, leading to a composite prediction. FST-based LTS has been studied by Luk and Damper [1996] and Sejnowski and Rosenberg [1987].

Joint-multigram model Both decision trees and HMMs make some independence assumptions among graphemes and/or phonemes, which simplify the model structure on one hand but might be too strong on the other hand. Ma and Randolph [2001] tried to solve this problem by testing various probabilistic structures based on Bayesian networks. The joint-multigram model proposed by Deligne et al. [1995] provided another approach toward a general structure.

A joint-multigram $u_i = \{\tilde{g}_i, \tilde{q}_i\}$ is a compound unit consisting of a grapheme component $\tilde{g}_i$ and a phoneme component $\tilde{q}_i$. The probability distribution over $U = (u_1, u_2, u_3, ..)$ is called a joint-multigram model, denoted as $P(U) = P(u_1, u_2, u_3, ...)$. In a joint-multigram model, the probability $P(u_i)$ is a grapheme-phoneme joint probability, and the probabilistic dependence among u_i is modelled by conventional n-gram models. With the joint-multigram model, no explicit probabilistic independence is assumed, and the dependence among graphemes and phonemes being modelled is limited only by the order of the n-gram model[1].

The joint-multigram model was applied to LTS by Bisani and Ney [2002] for English and German, where u_i was called a *graphone*. Vozila et al. [2003] implemented another variant of this model: they first generated n:1 grapheme-phoneme mappings using HMMs, and then collected all the grapheme-phoneme correspon-

[1] In the seminal paper of Deligne et al. [1995], independence is assumed among joint-multigrams, which is actually a special case of the n-gram model, i.e., a unigram model.

dences which they called *graphonemes*. Galescu and Allen [2002] built a bi-directional graphone-based system to perform both grapheme-to-phoneme (GTP) and phoneme-to-grapheme (PTG) conversion. Chen [2003] built a joint n-gram model by maximising the entropy of all training exemplars; they compared the joint n-gram model and conditional probability-based models, and concluded that the joint n-gram model accomplished better performance. Recently, Bisani and Ney [2008] discussed various issues when building joint-multigram models. They compared their implementation of the joint-multigram model and other LTS models, and reported that the joint-multigram model consistently outperformed other models on the LTS task.

We adopted the joint-multigram model in our work to predict pronunciations for OOV terms. The mathematical representation and practical implementation will be discussed in Chapter 4.

2.3.2.3 Comparison of different LTS approaches

Damper et al. [1998] compared the rule-based approach, analogy-based reasoning, ANN and IB1-IG based approaches, with the dictionary Teacher's Word Book (TWB) [Thorndike and Lorge, 1944]. They concluded that analogy-based reasoning outperformed others. Wolters and van den Bosch [1997] conducted a comparative study for various model-based approaches, including ANN, IB1-IG and IGTree, with Scottish Gaelic words [Wolters, 1997], and found that the IB1-IG and IGTree based approaches were the best. A similar comparison was conducted by Daelemans and van den Bosch [1996] on Dutch, in which they observed that an IGTree outperformed both an ANN and linguistic rules.

Shavlik et al. [1991] compared ID3 and ANN -based approaches on 5 data sets, and concluded that the ANN systematically outperformed the ID3 tree. The same comparison was performed by Häkkinen et al. [2003] on the CMU dictionary and MSU dictionary, which showed that the decision tree-based approach worked well on training data, however was worse than the ANN-based approach on held-out test data. The same conclusion was reached by Dietterich et al. [1995] and Baykal and Tolun [1998]. Dietterich et al. [1995] argued that ANNs outperformed decision trees because they could capture more statistical information; Baykal and Tolun [1998] reported similar results but attributed the superiority of the ANN-based approach to its robustness against noise and missing data.

Pagel et al. [1998] compared ID3 trees and CARTs, and got similar accuracy with the two approaches on dictionaries OALD, CMU and BRULEX; Andersen et al. [1996]

compared linguistic rules, CARTs and Tries on the corpora NetTalk, CMU, and ONOMASTICA, concluding that the Trie model worked as well as the rule-based approach, and the CART-based approach got the best result.

2.4 Uncertainty treatment

Uncertainty is ubiquitous in human speech and is the main challenge to all speech systems. For STD, uncertainty arises mainly from two sources: speech recognition errors and pronunciation variation. ASR errors might be caused by inaccurate speech models, environmental noise, pruning in decoding, OOV words, etc. Pronunciation variation might be caused by intra-speaker factors such as speaking speed, speaking style, emotional status, etc., and inter-speaker factors such as gender, age, accent [Mohamed et al., 2006]. All of these variations are combined and mixed, making them difficult to handle [Cucchiarini et al., 2000].

Three approaches have been developed to treat the uncertainty in STD: lattice-based representation, query expansion and soft match. The lattice-based approach has been described in Section 2.1.2; here we discuss query expansion and soft match.

2.4.1 Query expansion

The basic idea of query expansion is to constitute multiple search forms for a query, so that ASR errors and/or pronunciation variants can be explicitly represented. This approach has been extensively studied in speech transcription, for which the most popular approach is to create a multi-entry lexicon, either manually designed or automatically learnt from data [Sloboda and Waibel, 1996; Fukada and Sagisaka, 1997; Beulen et al., 1998; Ramabhadran et al., 1998; Kessens et al., 1999; Cremelie and Martens, 1999; Riley et al., 1999; Saraclar et al., 1999; Zhang et al., 2006; Turunen and Kurimo, 2006; Oh et al., 2007]. Another way to represent the alternative pronunciations is to build a stochastic mapping between the surface phones and the underlying HMMs, which is known as the hidden sequence model (HSM), proposed by Hain and Woodland [1999]; Hain [2001, 2002].

The idea of considering multiple pronunciations of a search term can be traced back to Christiansen and Rushforth [1977] who utilised multiple templates in DTW-based keyword spotting, and has been applied to SDR for some time; however, this approach has never been fully studied in STD until very recently [Mamou and Ramabhadran,

2008; Mamou et al., 2008; Can et al., 2009; Wang et al., 2009a]. The following review involves related work on keyword spotting, SDR and STD.

Acoustic expansion

This approach expands a search term by involving those terms with similar pronunciations. In practice, the search term is first converted to its pronunciation in the form of phoneme sequences, and then similar terms are found by examining the acoustic similarity. The acoustic similarity of two pronunciations is usually derived from the similarity of phoneme pairs according to a confusion matrix. Ng [1998] first proposed this approach in a keyword spotting system, and Logan et al. [2005] applied this approach to map OOV terms to INV phrases in SDR.

Phonetic expansion

Different from acoustic expansion, phonetic expansion extends a search term under phonetic constraints. This approach is a major contribution of this thesis, and had not been studied when we started our work. Very recently, IBM [Mamou et al., 2008; Can et al., 2009] and JHU [Sproat et al., 2008] presented their work on this approach, based on joint maximum entropy N-gram models, which is similar to our work that is based on joint-multigram models, as presented in Chapter 5.

Relevant expansion

In SDR, the results of a query might contain some 'representative terms' which are not in the original query but are very informative. These terms constitute a *relevant expansion* of the original query. This technique usually boosts the performance of a practical retrieval system. For example, Ng [2000] presented a 'relevant feedback' approach which refined a query by adding some terms in relevant documents and deleting some terms in irrelevant documents. Woodland et al. [2000] provided a similar approach, but expanded queries by adding terms from some parallel documents.

The same technique was also used to adapt the vocabulary and/or language model of a speech system [Kemp and Waibel, 1998; Aronowitz, 2008], which provided an easy way to migrate a practical system to new domains automatically.

In addition, Hu et al. [2004] expanded queries by adding semantically similar terms. For example, 'cold' was expanded by 'cool', 'study' was expanded by 'learn'. This can be regarded as a particular form of relevant expansion.

2.4.2 Soft match

A widely used approach to uncertainty compensation in keyword spotting and STD is known as *soft match*. Different from query expansion which constructs similar query terms, soft match allows a degree of mismatch between the canonical pronunciation and the detected pronunciation when searching for a term. With different measurements of the mismatch cost, this approach can be implemented in one of three ways.

Edit distance-based soft match

Distance-based soft match, sometimes called the minimum edit distance (MED)-based approach [Thambiratnam and Sridharan, 2005], derives the mismatch cost from the edit distance between the lexical form and detected form of the search term, with penalties for substitutions/insertions/deletions set empirically [James and Young, 1994; Szöke et al., 2005a; Miller et al., 2007; Mamou et al., 2008], or according to substitution rules [Thambiratnam and Sridharan, 2005], or targeting a minimum in the Bayesian risk (MBR) [Bosch et al., 2006] or the Bayesian distance [Amir et al., 2001].

Acoustic confusion-based soft match

In this method, the mismatch cost is derived from a confusion matrix that is constructed by running a phoneme recogniser on the training set and then aligning the recognition result with the reference transcript [Ng, 1998; Wechsler et al., 1998; Ng, 2000; Srinivasan and Petkovic, 2000; Audhkhasi and Verma, 2007; Amir et al., 2001; Pinto et al., 2008; Wallace et al., 2009]. Chaudhari et al. [2008] used a higher-order confusion matrix, which considers long-span phoneme contexts when evaluating phoneme pair confusion. Ramabhadran et al. [2009] proposed use of neural network instead of transcript forced alignment to derive the confusion matrix.

Model distance-based soft match

The acoustic confusion of two phonemes can also be computed from the similarity of their HMMs. Itoh et al. [2006] formulated this similarity based on the Bhattacharya distance of two Gaussian distributions; Iwata et al. [2008] formulated this similarity based on Kullback-Leibler divergence (KLD)

2.5 Confidence estimation

Confidence estimation plays an important role for STD in determining the reliability of putative detections and filtering out false detections. The following review summarises various approaches to confidence measurement. We concentrate on STD, but also look at some work on speech transcription. Reviews on this subject can also be found in [Siu and Gish, 1999; Wessel et al., 2001; Jiang, 2005].

2.5.1 Feature-based confidence

The first approach to estimating confidence is based on some 'features' that are generated during recognition, e.g., acoustic likelihood, language model scores, etc. Letting $c_1, c_2, ...c_3$ denote various features of a detection, this approach can be formally written as:

$$c = f(c_1, c_2, ..) \tag{2.1}$$

where c denotes the confidence score, and f denotes a mapping function which might simply be a selection, an interpolation, or a normalisation.

Various features have been studied. For example, Rohlicek et al. [1989b] proposed using duration-normalised acoustic likelihood, Cox and Rose [1996] studied second-phoneme-recognition normalised acoustic likelihood, Bergen and Ward [1997] used senone-score-normalised acoustic likelihood.

Kemp and Schaaf [1997] proposed to use various statistics from lattices, such as link probability, acoustic stability or hypothesis density. Manos and Zue [1997] studied and combined 5 features in a segment-based system, such as segment phonemic match score, lexical weight, etc.

Chase [1997] compared various features derived from decoding, such as the content of the N-best list, language model score, word pronunciation, word frequency in acoustic training materials, and separate-phoneme-recognition score. Decision trees, general linear models (GLMs), generalised additive models (GAMs) and neural networks were used to combine these features. Gillick et al. [1997] conducted similar experiments, considering features such as word duration, language model score, acoustic score minus the best score, n-best score, active node count, etc.; a generalised liner model (GLM) was applied for feature combination. Zhang and Rudnicky [2001] studied more features including acoustic features, language model features, word lattice features, N-best features, etc.

All of this research confirmed that the features derived from decoding, plus suitable normalisation and combination, can form a good measure for the confidence of a recognition hypothesis or a putative detection of a spoken term.

2.5.2 Posterior probability-based confidence

Let $P(K|O)$ denote the posterior probability that a speech segment O contains a spoken term K. According to Bayesian decision theory, a decision based on this posterior probability gives minimum risk when determining which term the speech contains. Therefore, the posterior probability is an ideal measure of the confidence of a detection. In practice, the Bayesian formula is usually applied to decompose the posterior probability into a ratio of the likelihood of the detected term and the evidence of the speech segment given the model, so we have,

$$\begin{aligned} c &= P(K|O) && (2.2)\\ &= \frac{p(O,K)}{p(O)} && (2.3)\\ &= \frac{p(O|K)P(K)}{\sum_{K'} p(O|K')P(K')} && (2.4) \end{aligned}$$

where K denotes the detected term, and K' represents any term that the speech segment may contain[2].

Equation 2.4 can be regarded as a normalisation that amends the likelihood-based confidence to make it comparable across utterances. This normalisation can be realised as a background model, a n-best list or a lattice.

Background normalisation

Background normalisation employs a background model to compute the model evidence $p(O)$ in Equation 2.3. This approach is formulated as the following equation,

$$c = \frac{S_{KW}}{S_{BA}} \quad (2.5)$$

[2]In this thesis, we use the capital P to denote the probability of a discrete variable or the joint probability of a group of discrete variables, and a lower-case p to denote the probability density of a continuous variable or the joint probability density of a group of variables in which at least one is continuous.

where S_{KW} is the likelihood with the keyword model, and S_{BA} is the likelihood with the background model. This approach was first proposed by Rose and Paul [1990], and was followed by James [1996].

N-best based confidence

Instead of normalising with an explicit background model, Rohlicek et al. [1989b] and Jeanrenaud et al. [1993] proposed a new method that normalised the likelihood of a frame at a certain state by the sum of the likelihood of that frame at all states, formally written as

$$c = \frac{p(s_t = e_K, O)}{\sum_{s'} p(s_t = s', O)} \tag{2.6}$$

where s_t denotes the state of frame t, e_K is the final state of K, and s' denotes any possible state at t. The joint probability $p(s_t, O)$ represents the probability that the speech frame at t resides in state s, usually computed with the Baum-Welch algorithm [Baum and Petrie, 1966].

Weintraub [1995] presented an n-best based confidence measure for LVCSR-based keyword spotting. In this method, the acoustic scores of the hypotheses in the n-best list involving the keyword are accumulated, and then are divided by the summation of the scores of all the n-best hypotheses. This is written as

$$c = \frac{\sum_{W:K\in\{w_i\}} P((w_1, w_2, ..., w_m)|O)}{\sum_{W} P((w_1, w_2, ..., w_m)|O)} \tag{2.7}$$

where $W = (w_1, w_2, ...w_m)$ is a hypothesis in the n-best list. Note that this measurement is not a likelihood ratio-based confidence, of the type that will be discussed in the Section 2.5.3, because the denominator here represents a smoothing item, instead of an alternative hypothesis. Jeanrenaud et al. [1995] presented the same measurement.

Setlur et al. [1996] simplified the n-best approach into a likelihood ratio of the best and the second-best hypothesis, formulated as

$$c = \frac{l(K_{best})}{l(K_{second})} \tag{2.8}$$

where $l(K)$ denotes the likelihood of term K, and K_{best} and K_{second} denote the best and next-best hypotheses in the n-best list. Note that this approach is a special case of the n-best normalisation,

$$c = \frac{l(K_{best})}{\sum_{i=1}^{N} l(K_i)} \tag{2.9}$$

where N denotes the size of the n-best list, and K_i is the i-th hypothesis.

Rueber [1997] investigated the relationship between the n-best normalised confidence and the correctness of the detection, and confirmed that they are indeed closely-related.

Lattice-based confidence

The n-best based confidence was extended to a lattice-based confidence by Wessel et al. [1998], in which the confidence of a detection is expressed as the ratio of the score accumulated over all complete paths passing the arcs of the detection to the score accumulated over all complete paths in the lattice, formulated as

$$c = \frac{\sum_{\alpha,\beta} l(\alpha,\beta,K)}{\sum_{\xi} l(\xi)} \tag{2.10}$$

where α and β denote any partial paths before and after the detected keyword K respectively, and ξ is any complete path in the lattice. Note that this formula is a special case of Equation 2.4 where $p(O,K)$ is approximated by an accumulated score of all paths involving K in the lattice.

Wessel et al. [1999] compared the n-best and the lattice -based confidence, concluding that the lattice-based confidence is superior. Because of the theoretical consistency with minimum decision risk and the practical computational efficiency, the lattice-based confidence has been widely adopted in keyword spotting and STD, e.g., [Woodland, 2000; Szöke et al., 2005a; Akbacak et al., 2008].

Aggregated posterior confidence

All of the above approaches derive the posterior probability $P(K|O)$ at the term level. Rivlin et al. [1996] proposed another approach, which first calculates the phone posterior probability of each frame, and then aggregates these frame-level posterior probabilities into a term-level confidence score. In [Rivlin et al., 1996], the frame-level phone posterior probability was calculated from phone class-conditional probabilities using the Bayesian formula. Mathematically, this approach can be expressed as

$$c = \prod_{t=t_{start}}^{t_{end}} P(q_t|O) \tag{2.11}$$

$$= \prod_{t=t_{start}}^{t_{end}} \frac{p(O|q_t)P(q_t)}{\sum_{q'} P(q'_t|O)P(q'_t)} \tag{2.12}$$

$$\tag{2.13}$$

where q_t denotes the phone label of the detected term at any time t between the starting time t_{start} and the ending time t_{end}, and q'_t denotes any valid phone in the phone inventory at time t.

The aggregation approach was followed by Bernardis and Bourlard [1998] in the HMM/ANN hybrid framework. Instead of applying the Bayesian formula, they calculated the frame-level phone posterior probabilities through a neural network. This approach was followed by a number of researchers, e.g., [Williams and Renals, 1999; Silaghi and Bourlard, 1999; Ketabdar et al., 2006].

2.5.3 Likelihood ratio-based confidence

Another confidence estimation approach is based on hypothesis testing. In this approach, the hit/FA decision is cast to testing the null hypothesis that 'the detected term is K' versus the alternative hypothesis that 'the detected term is not K'. Theoretically, this testing can be conducted by putting a threshold on the likelihood ratio of the null hypothesis and the alternative hypothesis, that is why we call the confidence derived using this approach as *likelihood ratio-based confidence*

Anti-word normalisation

The likelihood ratio-based confidence measuring was first proposed as a normalisation based on *anti-word models* [Gish et al., 1992; Rahim et al., 1995], written as

$$c = \frac{p(O|K)}{p(O|\bar{K})} \tag{2.14}$$

where $p(O|\bar{K})$ denotes the likelihood of term K with the anti-word model.

Rivlin [1995] proposed a similar approach, but based on the ratio of posterior probabilities instead of likelihood, such that,

$$c = \frac{P(K|O)}{P(\bar{K}|O)}. \tag{2.15}$$

Hypothesis testing

AT&T first stated the idea of casting confidence estimation to hypothesis testing [Rahim et al., 1995; Rose et al., 1995; Kamppari and Hazen, 2000]. This idea is formulated as follows,

$$c = \frac{p(O|K_{correct})}{p(O|K_{incorrect})} \tag{2.16}$$

where $K_{correct}$ and $K_{incorrect}$ represent the null and alternative hypothesis respectively.

This approach was first applied to the task of utterance verification (UV) [Rahim et al., 1997], in which the null hypothesis is 'the detected keyword really exists in the utterance' and the alternative hypothesis is 'this keyword is not actually present in the utterance'. In implementation, discriminative training is widely used, targeting to a minimum classification error rate (MCE) [Rahim et al., 1995, 1997] or a minimum verification error rate (MVE) [Sukkar et al., 1996; Sukkar and Lee, 1996; Setlur et al., 1996; Sukkar, 1998].

2.5.4 Discriminative confidence

In this approach, a hit/FA decision is cast to a binary classification. According to decision theory, an optimal classification strategy should base on *classification* posterior probabilities, instead of term posterior probabilities discussed in Section 2.5.2. This leads to the discriminative confidence formulated as follows,

$$c = P(C_{hit}|O,K) \tag{2.17}$$

where C_{hit} denotes the event that the detection of term K is a hit, and $P(C_{hit}|O,K)$ represents the probability that this detection is a correct one, given speech O.

Two-class modelling

The first implementation of the discriminative confidence estimation is a *two-class* approach, which models the class-conditional confidence distributions for correct and incorrect detections, and then derives the discriminative confidence using the Bayesian formula. Young [1994] proposed such an approach by computing correctness of the detections with various acoustic scores. Jeanrenaud et al. [1995]; Junkawitsch et al. [1996] modelled class-conditional probability density functions (cpfd) and derived from them term-specific thresholds. Cox and Rose [1996] implemented the two-class

approach by considering more features such as duration, the number of phonemes and the number of alternative hypotheses. Fetter et al. [1996] proposed the same approach, and investigated the relationship between discriminative confidence and likelihood ratio.

Discriminative models

The two-class approach requires a model of class-conditional probability distributions and hence is a generative approach; instead, a discriminative approach can be used that builds a discriminative model to estimate the classification posterior probabilities directly.

Mathan and Miclet [1991] utilised a MLP to generate a discriminative confidence for extra speech rejection. Weintraub et al. [1997] and Vergyri et al. [2006] proposed the same MLP-based confidence to justify recognition hypotheses. Linear discriminative functions were studied by Sukkar and Wilpon [1993]; Gillick et al. [1997]; Kamppari and Hazen [2000]. Generalised linear models (GLM) and generalised additive model (GAM) were studied by hung Siu et al. [1997]. Decision trees were studied by Neti et al. [1997]; Hauptmann et al. [1998], and SVMs were studied by Zhang and Rudnicky [2001]; Sudoh et al. [2006]; Shafran et al. [2006].

In addition, decision trees, GLMs, GAMs and neural networks were compared Chase [1997]; linear classifiers and ANNs were compared by Schaaf and Kemp [1997]. Both researchers reported that ANNs outperformed other discriminative models. Ábrego [2000] compared linear discriminant functions, ANNs and fuzzy logic, and reported that the fuzzy logic accomplished the best result. Zhang and Rudnicky [2001] studied decision trees, neural networks and SVMs, and reported that the best model was SVM.

Instead of estimating confidence measures, discriminative models have also been applied to make decisions directly, e.g., linear discriminative functions and MLPs were applied to make decisions by Lleida et al. [1993]; Ma and Lee [2007].

A major contribution of this work is a discriminative decision strategy which is based on discriminative confidence measures. Compared with the research discussed above, our work focuses on term-dependent decision factors and combines with confidence normalisation which targets at an ATWV-oriented decision making.

2.5.5 Relationship of various confidence measures

The four confidence estimation approaches described above are closely-related, which makes them consistent and complementary.

First of all, we note that the basic idea of confidence estimation, no matter how complex the implementation, is to derive some quantities that are discriminative for correct and incorrect detections, from some 'raw' features. Therefore Equation 2.1 is a general expression for all confidence estimation methods. If the mapping function f is a simple combination, then we get feature-based confidence; if it is a normalisation with a background model or an anti-word model, then we get the posterior probability-based confidence or the likelihood-ratio based confidence; finally, if it is a probabilistic discriminative model, we get a discriminative confidence.

Second, the posterior probability-based confidence and the likelihood ratio-based confidence are both developed from confidence normalisation. The difference is that the former bases itself on decision theory and the latter is based on hypothesis testing. Jiang [2005] pointed out that the hypothesis testing-based approach might be superior because more powerful anti-word models can be applied.

Third, both the likelihood ratio-based confidence and the discriminative confidence treat the hit/FA decision as a two-class classification problem, however the former is derived from hypothesis testing and the latter is derived from discriminative modelling.

Chapter 3

Experimental background

In order to present the work within context, we introduce the experimental settings and the data profile in this chapter. First of all we describe the experimental framework, describing each component in detail, and then discuss the experimental configurations, such as the work domain, the definition of out-of-vocabulary terms and the system tuning strategy. Afterwards, we describe the data we used for model training, system optimisation and performance evaluation. Using the data and configurations, we developed our baseline systems whose implementation and results will be presented at the end of this chapter.

3.1 Experimental framework

This section presents the experimental framework of the study. We will first illustrate the whole system, and then present each component in turn.

3.1.1 Framework overview

In order to allow comparisons with other systems, using standard evaluation paradigms, we adopted the architecture defined by NIST [NIST, 2006]. According to this architecture, an STD system is made up of two subsystems: an *ASR subsystem* that transcribes input speech into intermediate representations, and a *STD subsystem* that discovers potential occurrences of enquiry terms from the intermediate representations. Because lattices have the advantage of representing recognition alternatives and have been found to give better performance than single-best transcripts, we take lattices as the only intermediate representation in our study.

The STD subsystem can be further divided into a searching part called the *term detector* and a testing part named *decision maker*. Potential occurrences are found and assigned a confidence score by the term detector, giving a set of putative detections; then the decision maker examines the confidence score of each putative detection and determines if it is reliable enough to be an ascertained detection. Afterwards, the ascertained detections are output, and collected by a tool provided by NIST, which compares the detected and the true occurrences, and reports the detection quality measured by various evaluation metrics that will be discussed shortly.

The diagram of the entire framework is shown in Figure 3.1, for which we will describe each component in turn in the following sections.

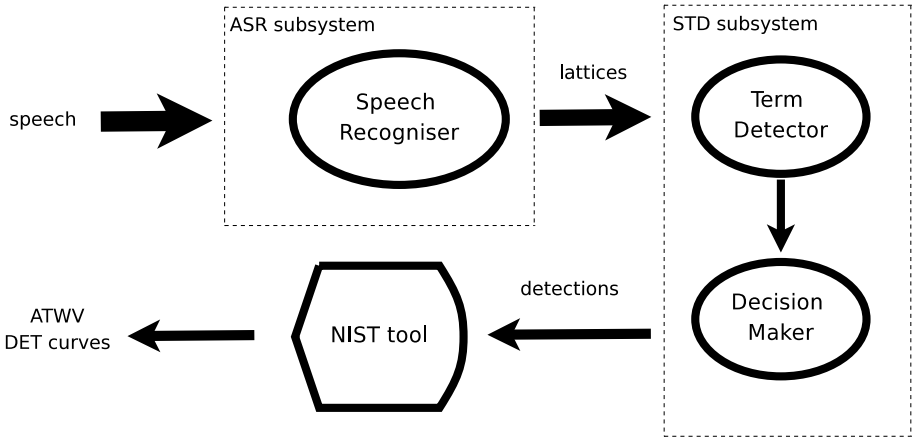

Figure 3.1: The experimental framework for STD. A speech recogniser constitutes the ASR subsystem. A term detector plus a decision maker constitute the STD subsystem. STD performance is evaluated by the NIST tool, with respect to ATWV and DET curve as introduced in Section 3.1.4.

3.1.2 ASR subsystem

The ASR subsystem is responsible for transcribing speech into word or phoneme lattices. This section gives a very brief review on the principal ideas and basic techniques which are used in a speech recognition system based on hidden Markov models.

HMM-based speech modelling

The Hidden Markov Model (HMM) is a powerful model for representing dynamic and statistic properties of a stochastic process, and hence is suitable to model speech signals which are highly random, varying, noisy and dynamic [Rabiner, 1989].

A HMM represents a probabilistic structure which contains a sequence of discrete latent variables following the first-order Markov assumption, and each latent variable *solely* determines an observation variable following certain probabilistic distribution. In speech recognition, adjacent latent variables with identical probabilistic properties are merged as a *state*, and each state emits a number of observations, say, speech frames here, subject to conditional independence. An example of this structure is depicted in Figure 3.2, which shows a 3-state HMM whose emissions follow a Gaussian mixture distribution.

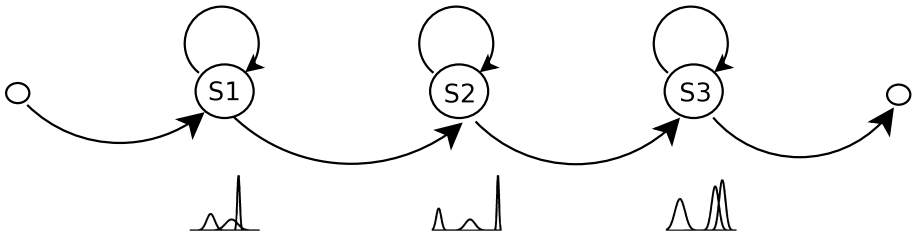

Figure 3.2: A typical HMM with 3 states and emissions following a Gaussian mixture distribution. Initial and ending states are added to facilitate model concatenation as discussed shortly.

Given a HMM denoted as H, the probability that speech O can be generated by H is written as Equation 3.3, where ξ denotes a state alignment with the speech, called a *state path*.

$$p(O|H) = \sum_{\xi} p(O,\xi|H) \tag{3.1}$$

Considering the fact that the emission probability of a frame is solely determined by the state issuing the frame, and applying conditional independence among frames, the probability of the entire speech is factorised into probabilities of each frame conditioned on the state at the frame-time. Furthermore, applying the Markov assumption, a state conditioning on historical states is simplified as conditioning on the previous state. This gives rise to Equation 3.3-3.5.

$$
\begin{aligned}
p(O|H) &= \sum_{\xi} p(O,\xi|H) && (3.2)\\
&= \sum_{\xi} p(O|\xi,H)P(\xi|H) && (3.3)\\
&= \sum_{\xi} P(\xi|H)\prod_{t} p(o_t|s_t,H) && (3.4)\\
&= \sum_{\xi}\prod_{t} p(o_t|s_t,H)P(s_t|s_{t-1},H) && (3.5)
\end{aligned}
$$

where o_t denotes the speech frame at time t, and s_t is the state in the state path ξ at time t. $p(s_t|s_{t-1})$ is the transition probability which is usually given by a transition matrix, and $p(o_t|s_t)$ is the emission probability usually in the form of Gaussian mixtures, given by Equation 3.6

$$p(o_t|s_t) = \sum_{i} \lambda_i G(o_t|\mu_i,\Sigma_i) \tag{3.6}$$

where $G(o|\mu,\Sigma)$ denotes a multivariate Gaussian distribution with mean μ and covariance matrix Σ. λ_i is the mixture weight of the i-th Gaussian component.

Hierarchical speech modelling based on HMMs

Although a HMM can represent any speech unit, we followed the conventional approach to make it model a context-dependent (CD) phone, and then model bigger units such as words and sentences by concatenating the CD-phone HMMs. This forms a hierarchical modelling architecture shown in Figure 3.3.

In this architecture, a sentence is broken down into a sequence of words, and each word is mapped to a phoneme sequence, either by looking up a dictionary or a letter-to-sound conversion. Note that *phonemes* are lexical units, thus need to be mapped to pronunciation units, namely, *phones*. In most cases, this phoneme-to-phone mapping is as trivial as one-to-one, but it might also be complex, depending on the phonetic rules. Phones are further projected to CD phones by taking account of the phonetic contexts. In this work, we just considered the direct neighbours, leading to the well-known *triphones*.

Each triphone can be modelled by a HMM, however this direct modelling will suffer severe data sparsity if some triphones have inadequate training instances. To solve this problem, similar triphones are usually tied together via a decision tree by asking some phonetic questions, and then the tied triphones can be readily modelled

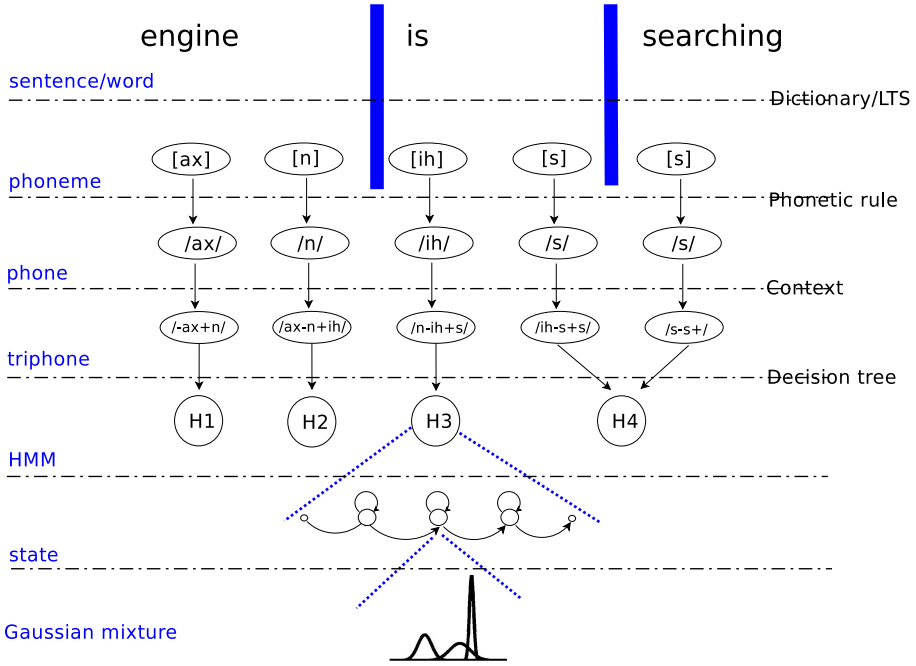

Figure 3.3: The hierarchical architecture of speech modelling based on HMMs. A sentence is decomposed into a sequence of words, and each word is mapped to a sequence of phonemes. Phonemes are then projected to phones by phonetic rules, and are further mapped to context-dependent phones (triphones here) by looking at the left and right neighbours. To deal with data sparsity, similar triphones (or states of similar triphones) are tied together and modelled by a single HMM.

by HMMs. The clustering technique can be also applied at the level of HMM states, as we did in this work.

Model training

Triphone HMMs can be trained using some transcribed speech, following a certain optimisation criterion, usually *maximum likelihood (ML)*. By ML training, the probability that the training speech are generated by the HMMs is maximised with respect to the HMM parameters, as formulated by the following equation,

$$\hat{\lambda} = \arg\max_{\lambda} p(O|H,\lambda) \tag{3.7}$$

where O denotes the training speech, H denotes the HMM sequence derived from the training transcript following the hierarchical architecture, and λ represents the model parameters, including state transition probabilities and parameters of Gaussian mixture distributions of each state.

The triphone HMMs described above represent the statistical properties of the speech in the acoustic space, and so are usually called *acoustic models (AM)*. Another important property of human speech is the dependency between words, which is critically important to hearing and understanding, for both human beings and computers. This dependency can be described by a joint probability distribution $P(W)$, where W denotes a sequence of words. The probability $P(W)$ is known as a *language model (LM)* since it models linguistic constraints on human languages.

Note that the language model can be built upon various linguistic units. We call a LM built on words a *word-based LM*, and a LM built on phonemes a *phoneme-based LM*, denoted as $P(Q)$ where Q represents a phoneme sequence.

In implementation, acoustic models can be trained efficiently using an Expectation-Maximisation (EM) algorithm [Huang et al., 2001]. For language models, the training may be as simple as counting the occurrences of word or phoneme fragments, yet it may be also quite complex when dealing with data sparsity, imbalance, and hierarchical constraints. We will present the AM and LM training strategies for the baseline system in Section 3.4, but investigating advanced training techniques is not within the main scope of this thesis.

Recognition

Taking the acoustic and language models, a speech recogniser can transcribe the input speech into sentences. Obviously, with a word-based LM, the recogniser generates word sequences, and with a phoneme-based LM, the recogniser generates phoneme sequences. We call an ASR system a *word-based ASR system* if it uses a word-based LM, and a *phoneme-based ASR system* if it uses a phoneme-based LM. The recognition task can be stated as a process of searching for a transcription that has the maximum posterior probability given the input speech, as formulated by Equation 3.8 and Equation 3.10 for word and phoneme -based ASR systems respectively, in which λ represents AM parameters, and $P(W)$ and $P(Q)$ are word and phoneme-based LMs. The optimal word and phoneme transcription, with respect to the posterior probability, are denoted $\hat{W}$ and $\hat{Q}$.

$$\hat{W} = \arg\max_{W} P(W|O,\lambda) \tag{3.8}$$

$$= \arg\max_{W} \frac{p(O|W,\lambda)}{p(O|\lambda)} P(W) \tag{3.9}$$

$$\hat{Q} = \arg\max_{Q} P(Q|O,\lambda) \tag{3.10}$$

$$= \arg\max_{Q} \frac{p(O|Q,\lambda)}{p(O|\lambda)} P(Q) \tag{3.11}$$

It is worth noting that within the hierarchical framework, the word sequence W and the phoneme sequence Q are decomposed to smaller units for acoustic modelling, so a word-based system and a phoneme-based system can share the same set of acoustic models but apply respective language models.

In implementation, a time synchronous decoding, called the Viterbi algorithm [Forney, 1973], is usually employed to conduct the recognition. We will present the experimental setup of decoding within the baseline system in Section 3.4.

Lattice generation

A lattice is a natural extension of a transcript $\hat{W}$ or $\hat{Q}$ which represents a single-best transcription. Within a lattice, alternative transcription results are merged and represented in a concise format, and rich information is retained in a compact structure. The retained information includes time stamps, acoustic likelihood, LM scores, pronunciation probabilities, etc. ; besides, more information can be derived from the lattice structure, e.g., lattice density and entropy, posterior probabilities, likelihood ratio. These derived quantities are widely used in the decision maker to ascertain putative detections in an STD task.

Obviously, word-based ASR systems generate lattices with word nodes, or *word lattices*, whilst phoneme-based ASR systems generate lattices with phoneme nodes, or *phoneme lattices*. An example of a phoneme lattice is shown in Figure 3.4.

In implementation, lattices are usually generated by allowing alternative partial paths retained in the decoding process discussed in the previous section, subject to some pruning criteria. The retained alternative results, plus associated information, are then arranged, merged and compacted into connected and acyclic graphs, stored in the specified format, giving a lattice.

3.1.3 STD subsystem

The STD subsystem searches lattices generated by the ASR subsystem for potential occurrence of enquiry terms, and examines each putative detection to determine if it is

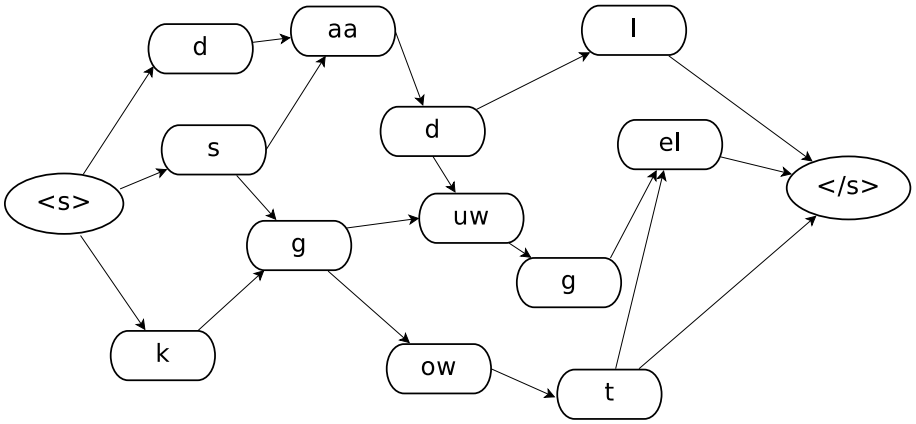

Figure 3.4: An example of a phoneme lattice.

reliable or not. This section describes the main work of the STD subsystem including search form conversion, lattice search, confidence estimation and decision making.

Word-based system and phoneme-based system

To begin with, we need a clear definition of word-based systems and phoneme-based systems. We have mentioned that an ASR system using a word-based LM is a word-based ASR system and generates word lattices, whereas if it uses a phoneme-based LM, it is a phoneme-based ASR system and generates phoneme lattices. We now define that for an STD system, if word lattices are used in the term search, it is a *word-based STD system*, and if phoneme lattices are used, it is a *phoneme-based STD system*. This leads to consistent definitions of ASR and STD systems regarding what type of system they are, therefore we can use the terms *word-based system* and *phoneme-based system* to refer an entire STD system or its subsystems without causing any confusion, because a word-based STD system can not have a phoneme-based ASR subsystem, and vice versa.

Search form conversion

An enquiry term can not be searched for before it is converted to a form that is compatible with the searched lattice. We define the compatible form of a term as its *search form*, and the task of finding this form as a *search form conversion*. For a word-based system, the search form is the term's word sequence, thus the conversion is trivial; for a phoneme-based system, however, the search form is the term's pronunciation, therefor the conversion could be a dictionary lookup if the term is an in-vocabulary (INV)

term, or a pronunciation prediction by employing some letter-to-sound (LTS) models if the term is out-of-vocabulary (OOV). In our experiments, a CART-based LTS model was used for the baseline system. Our extended work on LTS models, such as joint-multigram models and multiple pronunciation prediction, will be presented in Chapter 4.

Lattice search

Given the search forms, occurrences of enquiry terms can be searched for in the lattices, by matching the search forms to the partial paths in the lattices. To begin with, we need to arrange the search forms in an efficient way so that multiple terms and multiple search forms of the same term can be processed simultaneously. For that purpose, a dictionary tree is constructed such that each search form is represented by a full path from the root to a leaf, with the corresponding enquiry term labelled at that leaf. An example of a dictionary tree for phoneme-based search forms is shown in Figure 3.5.

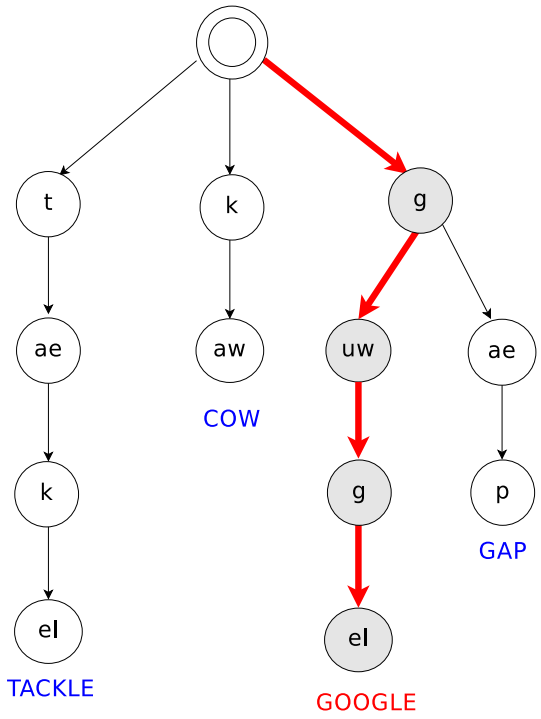

Figure 3.5: An example of a dictionary tree for phoneme-based search forms. Except the root node on the top, each node represents a phoneme. A search form is represented by a full path that goes from the root node to a leaf node, and the corresponding search term is labelled at the leaf. As an example, the shaded path in the tree represents the term 'GOOGLE'.

With the dictionary tree, lattice search is implemented as matching the complete paths in the tree to partial paths in the lattice. A recursive approach was adopted: for each node in the lattice, all the partial paths starting from that node are examined in a depth-first order and only those paths matching a partial path in the dictionary tree are retained and extended. If a leaf node of the dictionary tree is reached, the terms labelled to that leaf are detected. Figure 3.6 illustrates how the term 'GOOGLE' is detected from a phoneme lattice using this recursive matching.

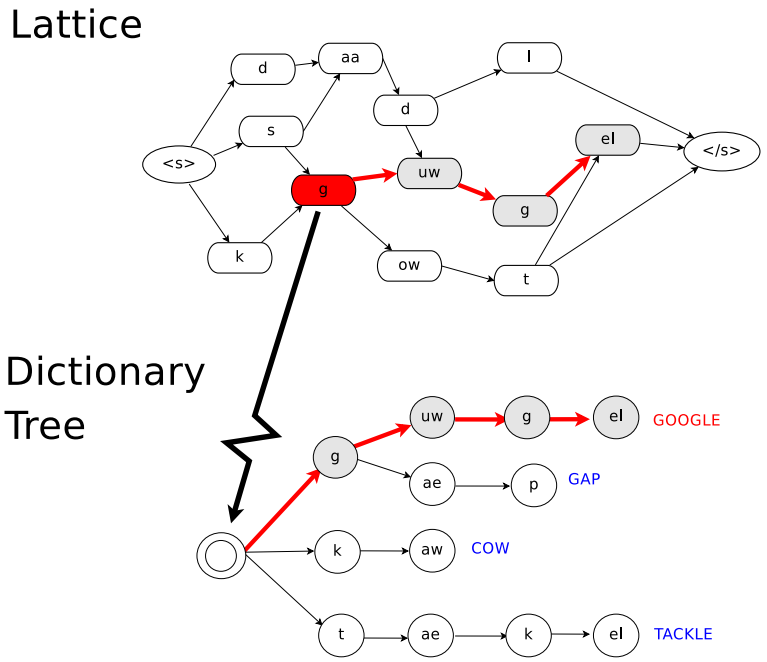

Figure 3.6: Lattice search demonstrated by searching for term 'GOOGLE' . When examining node g, all partial paths starting from g in the lattice are examined and only the paths that match a partial path in the dictionary tree are retained and extended. The matching-and-extending procedure is performed recursively, until the partial path $guwgel$ in the lattice is found matching the complete path that represents term 'GOOGLE' in the dictionary tree, indicating an occurrence of term 'GOOGLE'.

The complexity of the recursive algorithm discussed above can be written as $O(N \times (\beta \times B)^L)$ for a single term search, where N is the number of nodes of the lattice, B is the average number of fan-out arcs of these nodes, and L is the length of the phoneme sequence of the search term. Considering that only a portion of paths of the lattice can be extended after each phoneme comparison, a scale factor $0.0 < \beta < 1.0$ has been introduced in the formula, which represents the probability that a comparison gives a match.

When detecting a set of terms that are organised as a dictionary tree, the complexity is $O(N \times (\beta \times B \times D)^L)$, where D is the average fan-out of the dictionary tree, and L is the average depth of the tree. Note this may save much computation than searching every term individually, in which case the complexity is $O(M \times N \times (\beta \times B)^L)$ where M is the number of search terms.

In lattice search, we usually require that the phonemes corresponding to the partial path of the lattice match the phonemes corresponding to the full path of the dictionary tree one-by-one, or *exact match*. This approach is efficient to compute and performs well in the case of a dense lattice [Szöke et al., 2005a]. In the case of a sparse lattice, however, a more comprehensive approach that allows some substitutions/insertions/deletions between search forms and lattice paths may be helpful. This error-tolerant searching approach is called *soft match*. Moreover, if the search terms hold a high degree of pronunciation variation, such as the case of OOV terms, only considering the canonical pronunciations would be not enough. In this case, we can expand the canonical pronunciations by adding some possible alternatives, which gives rise to a stochastic pronunciation modelling (SPM).

For the baseline system, we simply adopted the exact match, leaving the SPM to be investigated in Section 5.3 and soft match in Section 5.4.

Detections and confidence measurement

From the lattice search, a set of putative occurrences of the search terms are detected. We define a detection d as a tuple containing all information we have about this detection, giving

$$d = (K, s, v_a, v_l, ...) \tag{3.12}$$

where K is the detected search term, and v_a, v_l are the acoustic likelihood and language model score respectively, and s denotes the speech segment covered by the detected term, again defined as a tuple representing the starting and ending time, written as

$$s = (t_{start}, t_{end}). \tag{3.13}$$

Note that there may be more information available for the detection, e.g., pronunciation probabilities and soft match probabilities, which we denote as "..." in Equation 3.12.

With the putative detections obtained, we need to judge whether they are reliable or not, based on certain confidence scores. Estimating the confidence score of a detection d can be formally written as a mapping from the detection to a scale:

$$f : d \mapsto c(d) \tag{3.14}$$

or expanded to

$$f : (K, s, v_a, v_l, ...) \mapsto c(d) \tag{3.15}$$

where $c(d)$ is its *confidence measure*, or simply *confidence* of the detection d as defined in Equation 3.12.

There are several ways to test the confidence of a detection [Jiang, 2005]. In this work, the lattice-based posterior probability was used as the confidence measure for the baseline system. We will derive the formula for this confidence measure as follows.

Denoting the event that a search term K appears between time t_1 to t_2 as $K_{t_1}^{t_2}$, posterior probability $p(K_{t_1}^{t_2}|O)$ hence represents the confidence that event $K_{t_1}^{t_2}$ takes place given speech O. On the other hand, detection $d = (K, s = (t_1, t_2), ...)$ can be regarded as the event that " the detector catches the event $K_{t_1}^{t_2}$ ", therefore the confidence of event d can be approximated by the confidence of event $K_{t_1}^{t_2}$, giving

$$c(d) = P(K_{t_1}^{t_2}|O) \tag{3.16}$$

where d is given by

$$d = (K, s = (t_1, t_2), ..). \tag{3.17}$$

The posterior probability in 3.18 can be computed from the lattice, that is why we call it the *lattice-based posterior probability*. Correspondingly, the confidence $c(d)$ is called the *lattice-based confidence*. To discriminate the lattice-based confidence from other confidence measures, we denote it $c_{lattice}$, thus we have

$$c_{lattice}(d) = P(K_{t_1}^{t_2}|O) \tag{3.18}$$

To calculate the lattice-based posterior probability, we note that it can be evaluated as the ratio of the accumulated probability of all paths involving $K_{t_1}^{t_2}$ to the evidence of the lattice, i.e., accumulated probability of all paths in the lattice, thus we have

$$P(K_{t_1}^{t_2}|O) = \sum_{\alpha,\beta} P(K_\alpha, K_{t_1}^{t_2}, K_\beta|O) \tag{3.19}$$

$$= \sum_{\alpha,\beta} \frac{p(K_\alpha, K_{t_1}^{t_2}, K_\beta, O)}{p(O)} \tag{3.20}$$

$$= \sum_{\alpha,\beta} \frac{p(O|K_\alpha, K_{t_1}^{t_2}, K_\beta)}{p(O)} P(K_\alpha, K_{t_1}^{t_2}, K_\beta) \tag{3.21}$$

$$= \sum_{C_K} \frac{p(O|C_K, K_{t_1}^{t_2})}{p(O)} P(C_K, K_{t_1}^{t_2}) \tag{3.22}$$

where K_α and K_β denote any paths before and after $K_{t_1}^{t_2}$, with K_α starting at frame 1 and K_β ending at frame T. To avoid cluttering notations, K_α and K_β are merged into C_K in Equation 3.22, representing the context of $K_{t_1}^{t_2}$.

In Equation 3.22, the conditional probability $p(O|K_{t_1}^{t_2}, C_K)$ is the acoustic likelihood of the path $K_\alpha K_{t_1}^{t_2} K_\beta$, and the prior $P(K_{t_1}^{t_2}, C_K)$ is provided by language models. The denominator $p(O)$ is a constant. In implementation, the Baum-Welch algorithm is applied to make the computation of $p(O|K_{t_1}^{t_2}, C_K)$ efficient, therefore we denote the lattice-based confidence based on this formula as *Baum-Welch confidence*. A simple approximation is achieved by replacing the sum over C_K with the score of the best path, giving

$$P(K_{t_1}^{t_2}|O) \approx \frac{max_{C_K}\, p(O|K_{t_1}^{t_2}, C_K)}{max_\xi p(O|\xi)P(\xi)} P(K_{t_1}^{t_2}, C_K) \tag{3.23}$$

where ξ denotes any complete path in the lattice.

Since the Viterbi algorithm is usually used to implement the computation, we call the confidence based on Equation 3.23 *Viterbi confidence*. Figure 3.7 demonstrates the computation of Viterbi confidence, where a detection of term 'GOOGLE' is tested.

We adopted the lattice-based Baum-Welch confidence in the baseline system, as it performed slightly better than the Viterbi confidence in experiments. Our extended work on confidence estimation based on discriminative posterior probabilities will be presented in Chapter 7.

Decision making

Given the confidence, a putative detection can be ascertained as a reliable detection or a false alarm. This is called *decision making*, and performed by the *decision maker*

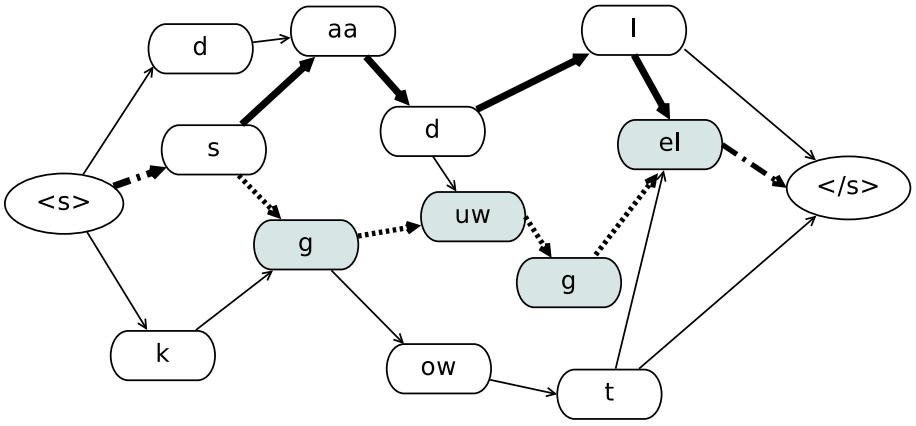

Figure 3.7: Computing the lattice-based Viterbi confidence for a detection of term 'GOOGLE'. The thick solid path is the best path through the lattice, and the thick dot path is the best path involving the pronunciation of 'GOOGLE'. The common parts of these two paths are represented by thick dot-dash lines. The lattice-based Viterbi confidence is the ratio of the scores of the thick dot path plus the dot-dash paths to the thick solid path plus the dot-dash paths.

in the STD architecture. Intuitively, the goal of the decision is to reduce as many as false alarms at the least cost of misses. Different applications require a different trade-off between false alarms and misses. For instance, in spoken document retrieval, we hope to find more relevant documents and so are willing to tolerate a number of false alarms; in spoken data mining, by contrast, precision is important because false alarms will deteriorate the decision quality. This FA/hit trace-off is controlled by a threshold θ in the decision maker, based on certain decision criteria.

A simple decision approach is to directly compare the confidence of a detection to the threshold θ, and assert it reliable, or an *ascertained detection* if and only if its confidence is equal to or higher than θ. This simple strategy can be formulated as a segmented decision function, written as

$$assert(d) = \begin{cases} 1 & if\ c(d) \geq \theta \\ 0 & if\ c(d) < \theta \end{cases} \tag{3.24}$$

where d is a detection for which the decision is to be made, and $c(d)$ is its confidence.

A notable property of this approach is that the threshold θ is term-independent, so is the decision strategy itself. More sophisticated approaches, such as term-dependent decision and discriminative decision will be studied in Chapter 5 and 6.

3.1.4 Evaluation

There are several metrics commonly used to evaluate performance of an STD system, and different metrics reflect different aspects of the performance. We first introduce these metrics, and then describe the evaluation process.

Figure of Merit (FOM)

A metric that has been widely adopted in keyword spotting is the *figure of merit (FOM)*, defined as the averaged detection rate over false alarms from 0 to 10 per hour, or roughly the detection rate with 5 false alarms per hour [Rohlicek et al., 1989b].

Let Δ denote the set of enquiry terms, and for each term $K \in \Delta$, let $H_K(m)$ represent the number of correct detections allowing m false alarms per hour. The FOM value of K is calculated as

$$FOM(K) = \frac{1}{11}\sum_{m=0}^{10} H_K(m) \tag{3.25}$$

$$\approx H_K(5) \tag{3.26}$$

The FOM value of term set Δ is calculated as the occurrence-average of the FOM values of all terms:

$$FOM(\Delta) = \frac{\sum_{K\in\Delta} N_{true}(K) FOM(K)}{\sum_{K\in\Delta} N_{true}(K)} \tag{3.27}$$

where $N_{true}(K)$ represents the number of true occurrences of term K.

The FOM metric possesses two properties: (1) Decision making is not required for calculating FOMs. Instead, all the putative detections are counted in the statistics $H_K(m)$ in Equation 3.25, and low-confidence detections are discarded automatically; (2) The overall performance $FOM(\Delta)$ is an 'occurrence average' of term-based performance $FOM(K)$, which makes the metric highly biased towards the performance on frequent terms.

It is also worth noting that the FOM value reflects an upper bound of the performance that an STD system can achieve with a perfect decision maker, because the computation spontaneously picks up a specified number of false alarms and throws other false alarms away, which equates to an ideal decision maker. The upper-bound performance is a good metric for research, however, it is not suitable to evaluate actual performance of a practical system for which a perfect decision maker is never feasible.

OCCurrence-weighted value (OCC)

NIST defines two new evaluation metrics in the STD evaluation plan [NIST, 2006]: *occurrence-weighted value (OCC)* and *average term-weighted value (ATWV)*. Both are designed to judge the practical performance of an STD system. We begin by describing the OCC value.

For a given set of terms Δ and some speech data, let $N_{hit}(K)$, $N_{FA}(K)$ and $N_{true}(K)$ represent the number of hits, false alarms, and true occurrences of term K respectively. In addition, the number of non-target terms (which gives the number of possibilities for incorrect detection) is represented as $N_{NT}(K)$. Then the miss and false alarm probabilities, $P_{miss}(K)$ and $P_{FA}(K)$ for each term $K \in \Delta$, are defined as follows:

$$P_{miss}(K) = 1 - \frac{N_{hit}(K)}{N_{true}(K)} \tag{3.28}$$

$$P_{FA}(K) = \frac{N_{FA}(K)}{N_{NT}(K)}. \tag{3.29}$$

In order to tune the metrics to give a desired balance of precision versus recall, a cost C_{FA} for false alarms is defined, along with a constant V for correct detections.

The occurrence-weighted value is computed by accumulating a value for each correct detection and subtracting a cost for false alarms as follows:

$$OCC = \frac{\sum_{K\in\Delta}[VN_{hit}(K) - C_{FA}N_{FA}(K)]}{\sum_{K\in\Delta}VN_{true}(K)}. \tag{3.30}$$

It can be seen that OCC and FOM values are similar in that they are all occurrence-weighted therefore both are susceptible to frequent terms. The difference is that false alarms can not be automatically discarded when calculating OCC values. This indicates that to get a high performance in terms of OCC values, it is important to design a reliable decision maker so that potential false alarms would be caught and discarded before going to evaluation.

Average Term-Weighted Value (ATWV)

Although the OCC value is a good indicator of practical performance, it is still an occurrence-weighted measure, thus is inherently biased toward frequent terms. To solve this problem, NIST introduced another metric, the average term-weighted value (ATWV), which is defined by averaging a weighted sum of miss and false alarm prob-

abilities, $P_{miss}(K)$ and $P_{FA}(K)$, over terms:

$$ATWV = 1 - \frac{\sum_{K \in \Delta} [P_{miss}(K) + \beta P_{FA}(K)]}{|\Delta|} \tag{3.31}$$

where $\beta = \frac{C}{V}(P_{prior}(K)^{-1} - 1)$, and $|\Delta|$ represents the size of Δ. The NIST evaluation tool provided for the NIST 2006 STD evaluation sets a uniform prior term probability $P_{prior}(K) = 10^{-4}$, and the ratio $\frac{C}{V}$ to be 0.1 with the effect that there is an emphasis placed on recall compared to precision in the ratio 10:1.

Similar to the OCC value, calculating ATWV does not discard any false alarms, and hence requires a carefully designed decision maker. Since it is term-weighted, all terms should be treated equally for a high ATWV.

Detection Error Tradeoff (DET) curve

A shortcoming of the metrics discussed above is that they all give a *point measurement* for STD performance, i.e., a single value that merges recall and precision. Although easy for reading and comparison, the point measurement misses the complete picture of the performance of an STD system with respect to the balance of hits and false alarms. To examine the performance more thoroughly, miss rates are plotted versus false alarm rates, leading to a *detection error tradeoff (DET) curve* [Martin et al., 1997].

In such a diagram, false alarm rates are put on the x-axis, and miss rates are put on the y-axis. Both axes are presented in a normal deviate scale so that the plotted curve is approximately linear. The ATWV is measured at a particular point on the DET curve, which corresponds to a particular confidence threshold that is usually specified by optimising the ATWV result on the development set. In many cases, the threshold tuned on the development set is not optimal for the evaluation set, and hence the ATWV is usually not the maximum ATWV along the DET curve. We denote the maximum ATWV along the DET curve the *max-ATWV*, which reflects the best performance of an STD system with an ideal confidence threshold. A typical DET curve is plotted in Figure 3.8, with the ATWV and max-ATWV marked as a cross and a star respectively.

Evaluation process

In this work, ATWV and DET curves are the major metrics we use to report experimental results, as they are standard metrics in the NIST evaluation. In addition, max-ATWV is also reported to assist performance analysis if necessary; and false-alarm

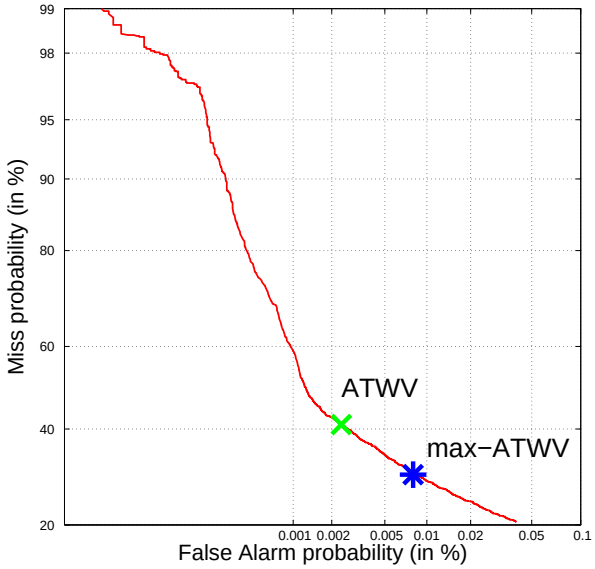

Figure 3.8: A typical DET curve with the ATWV and max-ATWV marked. The cross denotes the ATWV, and the star denotes the max-ATWV.

rates and miss rates may be reported as well to demonstrate system behaviour in more details.

When performing evaluation, ascertained detections produced by the STD system are fed into the NIST evaluation tool, which compares the hypothesised occurrences and the true occurrences given by a reference file and reports results in terms of ATWV and DET curves.

3.2 Experimental configurations

We present in this section the configurations of our experiments, especially the working domain, the accurate definition of out-of-vocabulary terms and the two-phase tuning strategy for system optimisation.

3.2.1 Experimental domain

We chose the meeting domain to conduct the experiments. Meeting is an important way for people to exchange knowledge and ideas. Therefore, meetings contain rich information that is highly valuable for a wide range of applications. With storage devices becoming cheaper, a huge amount of speech data are being archived from

meetings, conferences and lectures. There is a demand for information retrieval from such meeting speech archives.

Besides practical importance, the considerable challenges faced by ASR systems on meeting speech is the other reason we chose this domain. In meetings, the environment is usually noisy, participants tend to talk spontaneously, and topics are often diverse. These adverse conditions cause more recognition errors than on read speech, making term detection more challenging.

Finally, the meeting domain fits our interest in OOV terms. In meetings, people tend to talk about new techniques, new fashions and new business, which are much likely out-of-vocabulary. Furthermore, people in meetings tend to have little knowledge about the novel terms and their pronunciations, leading to some OOV specific phenomena, e.g., slow speaking, abnormal pitch, much pronunciation variation, etc. These OOV specialities are the subjects of our research.

For these reasons, our experiments were set up on meeting speech data; specifically, we chose the condition of individual headset microphones (IHM), which involves most of the spontaneous speech effect and less environmental noise compared to the distant microphone condition, making the phenomena we study more related to human pronunciation rather than acoustic noise.

3.2.2 Definition of out-of-vocabulary terms

The main goal of this study is to improve the performance of an STD system on out-of-vocabulary (OOV) terms; for that, we first of all need a clear definition of OOV words and OOV terms. To define out-of-vocabulary, however, we should first clarify what a *vocabulary* is.

ASR vocabulary and STD vocabulary

A vocabulary is a list of words/terms recognised by a speech system. Herein, that a word or term is 'recognised' means its pronunciation can be obtained from an associated dictionary. For STD, there are two vocabularies: one is for the ASR subsystem to generate word/phoneme lattices, which we call the *ASR vocabulary*; the other is for the STD subsystem to conduct term search, which we call the *STD vocabulary*. Accordingly, the dictionaries associated with the ASR and STD vocabulary are called the *ASR dictionary* and *STD dictionary* respectively. These two vocabularies (and the associated dictionaries) are fundamentally different not only in concept, but also in the

	Word-based system	Phoneme-based system
ASR dictionary	word:pron. (google:g uw g l)	phoneme:phoneme (g:g)
ASR vocabulary	word list(google)	phoneme list(g)
LM	word n-gram	phoneme n-gram
ASR lattice	word lattice	phoneme lattice
STD dictionary	word:word (google:google)	word:pron.(google:g uw g l)
STD vocabulary	word list (google)	term list (google)
Search form	word sequence(google)	phoneme sequence(g uw g l)

Table 3.1: Comparison of word-based and phoneme-based systems in terms of dictionaries, vocabularies, lattices, LMs and search forms . Examples are shown in brackets, and 'pron.' abbreviates 'pronunciation'.

content they have in a real system. For example, in a word-based system, the ASR dictionary contains a large number of words and their pronunciations, while the STD dictionary contains all words of the search terms, with the 'pronunciation' of a word simply being the word itself.

Now we can compare a word-based and a phoneme-based system, in terms of dictionaries, vocabularies, LMs, lattices and search forms, as shown in Table 3.1. As an example, a word 'google' is used to show what the entries of the vocabularies, dictionaries and search forms should look like.

The OOV issue for word-based and phoneme-based systems

With vocabularies defined, we can define OOV words for ASR and OOV terms for STD tasks. For the ASR subsystem, if a word is absent from the ASR dictionary, then it is an OOV word. For the STD subsystem, if a word is absent from the STD dictionary, then it is an OOV word, and any term that contains at least one OOV word is an OOV term.

According to these definitions, we can see that in ASR tasks, a word-based system will have an OOV issue if the ASR vocabulary is limited, yet a phoneme-based system does not have this problem since its vocabulary contains all phonemes. In STD tasks, on the contrary, a word-based system has no OOV problem since any term can be added to the STD vocabulary on the fly without any difficulties, while a phoneme-based STD system will meet an OOV term if any word of the term is unknown to the STD dictionary.

When we examine the OOV issue on the whole system, i.e., ASR plus STD, it becomes clear that for the word-based system, the OOV issue arises from the limited ASR dictionary, while for the phoneme-based system, the OOV issue arises from the limited STD dictionary. No matter where the OOV issue comes from, it can be remedied by augmenting the ASR or STD dictionary with OOV words whose pronunciations are predicted by some LTS models. However, this remedy does not work for word-based systems, since the remedy requires re-transcribing the speech, which is prohibited by the NIST architecture. This is why phoneme-based systems are widely used to deal with the OOV issue.

Strict definition of OOV words and terms

Although the definitions of OOV words for both ASR and STD tasks have been proposed, there is another concern in realistic experiments: are those OOV words allowed to appear in the training materials for acoustic models and language models? Some authors say 'yes', e.g., Akbacak et al. [2008]. However we are cautious to make that decision. The realistic scenario, to our mind, is that the novel words should not or rarely be used by people in the past. This means the OOV words are not only out-of-vocabulary of an ASR or STD system, but also out-of-vocabulary of a language. In practice, our study simulates this scenario, and thus prohibits OOV words from existing in training materials. More specifically, we define OOV words strictly as follows:

- Lexical layer: OOV words are those words absent from the ASR vocabulary and STD vocabulary;
- Acoustic layer: OOV words should not appear in the training data for the acoustic models;
- Linguistic layer: OOV words should not appear in the training data for the language models.

By this definition, to start our experiments, we first need a set of terms whose words do not exist in the training speech data and text but have instances in the evaluation data. However, there are only a small number of such *natural* OOV terms in our database. To get sufficient OOV instances for experiments, we took the following approach: first we manually select a set of terms which are suitable to be OOV terms and have some instances in the evaluation data, and then remove these terms from the

ASR and STD dictionary and the speech and text training data. We call this removing process *OOV-purging*.

3.2.3 Two-phase tuning

A practical problem when conducting STD experiments is how to choose optimal parameters for speech transcribing and term detection. For the ASR subsystem, the tunable parameters include the threshold for triphone clustering, the number of Gaussian mixtures, the LM scale factor, the word insertion penalty, etc; for the STD subsystem, the tunable parameters include the LM score, the word insertion penalty, the confidence threshold, etc. These parameters are optimised with respect to the performance of a development set.

The question is, should we tune the ASR and STD subsystems together or separately? Tuning together, or integrated tuning, obviously gives better performance (at least on the development set), whereas it might be not the best in practice. Firstly, the integrated tuning is inconsistent with the standard STD architecture in which the ASR and STD subsystem should operate separately. In fact, the STD subsystem is ideally able to detect terms from lattices generated by any ASR system without knowing details of the recogniser. Secondly, the STD subsystem usually needs to be optimised with fixed lattices. It may employ more powerful LMs, better LTS models, more robust confidence measures, more reliable decision strategies, etc.

For these reasons, we adopted the separate tuning approach in experiments, by which we first tune the ASR subsystem to maximise recognition accuracy and then tune the STD subsystem to maximise detection performance. We call this tuning approach *two-phase tuning*.

3.3 Data profile

In this section, we present the data resource used in this work. We first present how we selected the search terms, and then describe the speech and text corpora used for model training and evaluation. Most of the resources, including the speech and text corpora and vocabularies, came from the Augmented Multiparty Interaction (AMI) project, especially those used by the AMI RT05s LVCSR system which was developed for the Rich Text 2005 spring evaluation organised by NIST [Hain et al., 2006a] .

3.3.1 Terms for search

Search terms

According to NIST's definition, an STD system should handle both single-word and multiple-word terms such as 'Bush's new flight'. In this work, however, we just focus on single-word terms. This because the number of occurrences of multiple-word OOV terms is too small to make experiments on OOV terms possible. Therefore most of the OOV terms we selected are single words. We assume that techniques developed for single-word terms will readily apply to multiple-word terms.

We first selected 256 regular words or compound words as INV terms, all of which are content words occurring 4 to 20 times in the evaluation data. In addition, with the method discussed shortly, 484 OOV terms were selected for evaluation. From these OOV terms, 67 terms were selected for system tuning.

In summary, we defined three term lists: 67 OOV terms for system tuning; 484 OOV terms and 256 INV terms for performance evaluation.

Selecting OOV terms

To simulate the real scenario of detecting new terms using an existing STD system, we tried to find terms that are popular at present but were not present at some point in the past, which we call *real OOV terms*. Specifically, we compared the AMI dictionary to the COMLEX Syntax dictionary v3.1 which was published by LDC in 1996, and selected 412 terms in the AMI dictionary but not in the COMLEX dictionary. All these selected terms are give in [Wang, 2009]. Figure 3.9 shows the occurrence histogram of these real OOV terms in the evaluation set.

Besides these 412 real OOV terms, we further selected 70 *artificial OOV terms*. These terms are all nouns and suitable as search terms. Table 3.2 lists these terms, and Figure 3.10 shows the occurrence histogram.

Table 3.3 summarises the definitions of various OOV terms. Combining the real and artificial OOV terms, we get the final OOV term list, which consists of 482 content terms in total. The numbers of occurrences of these terms in the evaluation set are given in Table 3.4; and the occurrence histogram is shown in Figure 3.11.

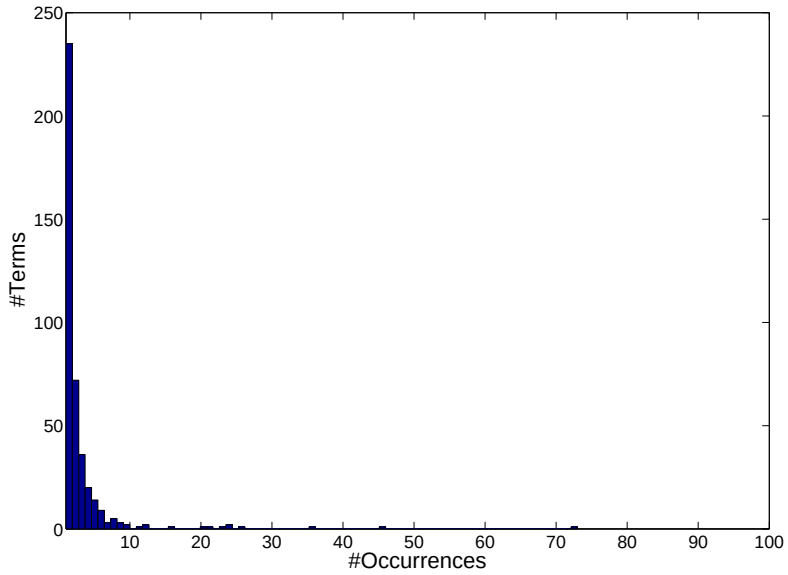

Figure 3.9: The occurrence histogram of the real OOV terms. The x-axis represents the number of term occurrences in the evaluation set, and the y-axis represents the number of terms with this number of occurrences.

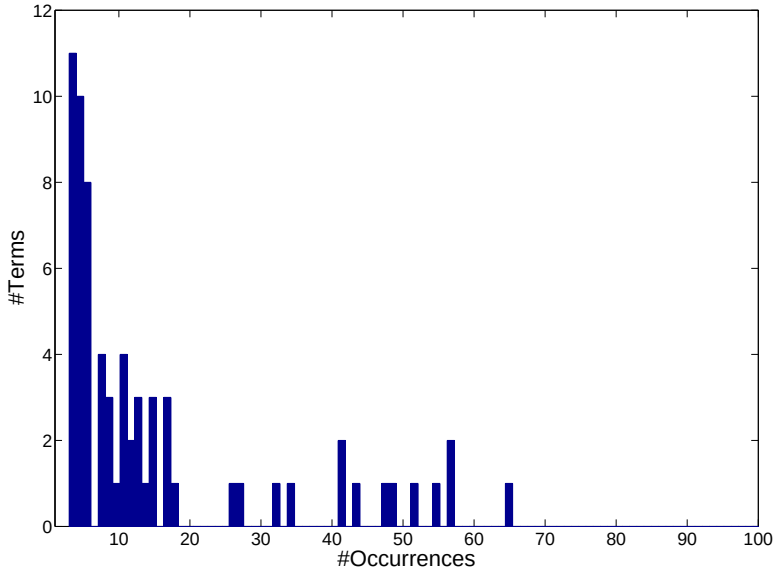

Figure 3.10: The occurrence histogram of the artificial OOV terms. The x-axis represents the number of term occurrences in the evaluation set, and the y-axis represents the number of terms with this number of occurrences.

AGENDA	ALIGNMENTS	AMBIGUITY	ARMY
BUDGET	CHUCK	COAST	COLLECTION
COMPUTER	CONTROL	COURSE	CRITERIA
DEFINE	DELTA	DIFFERENCE	DISTANCE
DOMAIN	FIFTY	FIGURE	GENDER
GENERATION	HORN	HOTEL	INFORMATION
JET	MEAT	PASSWORD	POSTER
LANGUAGE	LAPTOP	LINDER	MARKETING
MOUSE	NETWORK	NOISE	ORGANIZATION
PENALTY	POLLUTION	POPULATION	POSSIBILITY
PROGRAM	PROJECT	QUESTION	RECOGNITION
REMOTE	RESOLUTION	ROCK	SCHOOL
SECURITY	SIR	SPEECH	STANDARD
STRATEGY	SUMMER	SYSTEM	TARGET
TRANSCRIPTION	TROLLEY	TWENTY	UNDERGRADUATE
WEATHER	BROADCAST	COMPETITION	DESKTOP
SECRETARY	STATEMENT	TELEVISION	UTTERANCE
ELECTRONICS	REGION		

Table 3.2: List of the artificial OOV terms.

	Definition
Natural OOV terms	Terms that are invented in the history of language evolution and are absent from the system dictionary of an STD system.
Real OOV terms	Terms defined in this work that exist in the present AMI dictionary but are absent from the COMLEX Syntax dictionary 3.1 published in 1996.
Artificial OOV terms	Terms defined in this work that are intentionally removed from the present AMI dictionary, for the study of OOV phenomena.

Table 3.3: The definitions of natural OOV terms, real OOV terms and artificial OOV terms.

Corpora	rt04seval	rt05seval	ami08	rt04s+rt05s+ami08
Real OOV terms	177	177	107	412
Real OOV term occ.	328	348	467	1143
Artificial OOV terms	42	60	33	70
Artificial OOV term occ.	212	420	961	1593

Table 3.4: The occurrences of the selected OOV terms in the evaluation set. Note 'occ.' represents 'occurrence'.

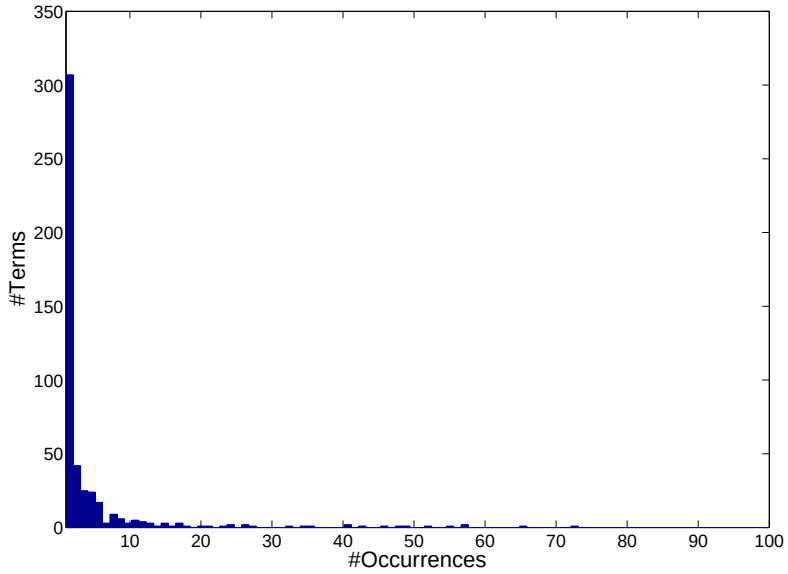

Figure 3.11: The occurrence histogram of all the selected OOV terms. The x-axis represents the number of term occurrences in the evaluation set, and the y-axis represents the number of terms with this number of occurrences.

3.3.2 Speech corpora

Data source

The speech data used in this work are from multi-participant meetings recorded in several institutes, including the International Computer Science Institute (ICSI), the National Institute for Standards and Technology (NIST), the Carnegie Mellon University Interactive Systems Laboratory (ISL), the Linguistic Data Consortium (LDC), the Virginia Polytechnic Institute and State University (VT) and partners of the AMI project. All the meetings were recorded with individual head-mounted microphones (IHM) and a set of multiple distant microphones (MDM), though we only used the

IHM recordings in this work.

All the recordings were manually transcribed into word transcripts. Using these transcripts, long audio files were cut into short segments, and pure silence was discarded.

Training set

The data for acoustic model training were recorded at four sites: 73 hours of speech from 30 technical meetings at ICSI [Janin et al., 2003], 13 hours of speech from 15 meetings at NIST, 10 hours of speech from 19 meetings at ISL [Burger et al., 2002], and 16 hours of speech from 35 meetings by AMI partners [Hain et al., 2006b]. In total, the training corpora contain 104 hours of speech with silence excluded. This training set is denoted by *icsinistislami05*. According to the strict OOV definition in Section 3.2.2, we purged OOV terms by deleting those sentences containing any OOV terms. 23% of the training data were removed by this purging, leaving 122744 utterances 80.2 hours of speech for the OOV-free AM training. This OOV-purged training set is denoted by *icsinistislami05-purged*

Development set

The data set used for system tuning is the official development set for the NIST RT04s evaluation, denoted as *rt04sdev*. It contains 1.40 hours of speech excerpted from 8 meetings recorded at ICSI, NIST, ISL and LDC.

Evaluation set

A large evaluation set was designed in order to obtain a reasonable coverage of OOV terms. This evaluation set, denoted as *rt04srt05sami08*, consists of three subsets: *rt04seval*, *rt05seval* and *ami08*.

- *rt04seval*: the official evaluation set for the NIST RT04s evaluation, containing 1.7 hours of speech excerpted from 8 meetings recorded at ICSI, NIST, ISL and LDC.
- *rt05seval*: the official evaluation set for the NIST RT05s evaluation, containing 2.1 hours of speech excerpted from 10 meetings recorded at ICSI, ISL, VT and AMI partners.

1	AMI008-ED1002a
2	AMI008-ED1002b
3	AMI008-ED1002c
4	AMI008-ED1003a
5	AMI008-ED1003b
6	AMI008-ED1003c
7	AMI008-ED1005a
8	AMI008-ED1005b
9	AMI008-ED1005c
10	AMI008-ED1007a
11	AMI008-ED1007b
12	AMI008-ED1007c

Table 3.5: The meetings we selected to use from the new recorded speech corpus *ami08*.

- *ami08*: a subset of the AMIDA meeting corpus, recorded at the University of Edinburgh in 2007-2008, containing 7.2 hours of speech from 12 meetings.

The subset *rt04seval* and *rt05seval* have been documented by NIST [2007], while *ami08* is relatively new. For reference, Table 3.5 lists the meetings in *ami08* that were used in this work.

As a summary, Table 3.6 lists all the speech corpora for this study.

3.3.3 Dictionary

We need a word dictionary for several purposes in this work:

1. Train/test the ASR subsystem;
2. Convert the text corpora from word text to phoneme text so that phoneme-based LMs can be trained;
3. Train LTS models to predict pronunciations for OOV terms.

We started from the dictionary used by the AMI RT05s system, named *AMI05s*. This dictionary covers 50002 frequent words collected using a procedure outlined in

	Training (utt./h.)		Dev (utt./h.)	Eval (utt./h.)		
				rt04srt05sami08		
	icsinistislami05	icsinistislami05-purged	rt04sdev	rt04seval	rt05seval	ami08
ICSI	101136/66.7	91200/54.5	509/0.35	604/0.42	596/0.44	-
NIST	11767/12.8	10553/10.0	377/0.35	560/0.40	638/0.41	-
ISL	10476/8.9	9515/6.7	366/0.32	694/0.45	749/0.46	-
LDC	-	506/0.38	0	643/0.40	-	-
AMI	13443/15.5	11476/9.0	-	-	570/0.39	5463/7.2
VT	-	-	-	-	577/0.35	-
SUM	136822/103.9	122744/80.2	1758/1.4	2501/1.7	3130/2.1	5463/7.2
TOTAL	136822/103.9	122744/80.2	1758/1.4	11094/11.0		

Table 3.6: The speech corpora used for this study, in which 'Training', 'Dev', 'Eval' denotes the training set, development set and evaluation set respectively, and 'utt.' and 'h.' denotes 'utterance' and 'hour' respectively.

[Hain et al., 2005]. Pronunciations are based on the UNISYN pronunciation lexicon [Fitt, 2000].

In order to perform experiments for OOV terms, we deleted the OOV terms from the AMI dictionary, resulting in an OOV-purged dictionary that contains 49620 words, named *AMI05s-purged*. This OOV-purged dictionary was used in the word-based system as the ASR dictionary to conduct speech transcribing, and was used in the phoneme-based system as the STD dictionary to provide pronunciations for INV terms. Furthermore, this dictionary was used to train LTS models to predict pronunciations for OOV terms. Table 3.7 lists these two dictionaries in the first two rows.

Besides the word dictionary, we also need a phoneme dictionary, named *AMI05s-phn*, for the phoneme-based ASR system to perform transcribing. In our experiments, the phoneme dictionary contains 44 non-silence phonemes, which is the same as the AMI RT50s system. *AMI05s-phn* is summarised in the third row of Table 3.7, and the entire phoneme list is presented in [Wang, 2009].

3.3.4 Text corpora

We trained our language models on the same text data used by the AMI RT05s system, except that the OOV terms were purged out. These training corpora, provided by Vincent Wan from the University of Sheffield, contain text from various sources such as news, transcripts of speech corpora, and a huge amount of web text collected from

Dictionary	#Words	#Entries	Comments
AMI05s	50002	50740	origin from AMI RT05s LVCSR system
AMI05s-purged	49620	50351	real and artificial OOV words purged
AMI05s-phn	47	47	phoneme set used by AMI05s

Table 3.7: The dictionaries used in this work. Note that the word dictionaries contain multiple pronunciations therefore the number of words and the number of entries are not equal. The phoneme dictionary contains 44 non-silence phonemes, 1 short pause, plus a starting silence and an ending silence of a sentence.

the Internet. This training set is denoted as *STDTXT*, consisting of 8 corpora as shown in Table 3.8.

To purge OOV terms, we tested 3 purging approaches:

1. sentence-deletion: delete the whole sentence if it contains OOV terms;
2. term-deletion: delete OOV terms only and keep the rest of the sentence;
3. count-deletion: delete counts of the OOV term from the intermediate statistics.

Table 3.9 shows the perplexities of the models trained based on these three purging approaches. To make the conclusion secure, we conducted the experiment on 4 subsets of the corpora. From the results we have a consistent conclusion, that sentence-deletion is the best way for OOV purging, which suggests that prohibiting incorrect contexts is more important than accumulating counts for correct word sequences in LM training.

Having found the best purging approach, we applied it to remove OOV terms from the training corpora *STDTXT*, giving the OOV-free training set, denoted as *STDTXT-purged* and reported on the bottom of Table 3.8. We see that 20% of the text data were removed in order to get rid of the OOV terms.

3.4 LVCSR baseline system

To generate high-quality lattices for STD, we first built a standalone LVCSR system, and then made use of it as the ASR subsystem for the STD baseline. We present the implementation of the LVCSR system in this section, and report the experimental results with respect to the NIST RT05s evaluation.

Corpus	#Words (MW)
Swbd/CHE	3.5
Fisher	20
ICSI/AMI	1.7
Web(Swbd)	166
Web(Fisher topics)	158
Web(ICSI)	132
Web(AMI)	104
Web(CHIL)	68
STDTXT	653.2
STDTXT-purged	521.4

Table 3.8: The text corpora used for language model training. *STDTXT* contains all the original text, while *STDTXT-purged* has OOV terms purged out. The second column reports the size of the corpora, in terms of million words.

3.4.1 System implementation

Implementation strategy

To ensure its quality, we built the LVCSR system in two steps: firstly, we duplicated the training and testing process of a state-of-the-art LVCSR system, such as, the AMI RT05s system, to build an imitative system; afterwards, we purged the OOV terms from AM and LM training materials and built an OOV-free LVCSR system. This OOV-free system is denoted as the *LVCSR baseline system*, which will serve the ASR subsystem for lattice generation.

The AMI RT05s system, whose diagram is shown in Figure 3.12, is rather complex. For our purpose, we just reproduced the P1 system that implemented a basic ASR framework without advanced training and decoding techniques. We assumed that the moderate recognition performance given by this basic system would be enough for the study on STD. In fact, the error-prone ASR results provided more challenging examination on the STD techniques we developed in the study.

In implementation, the HTK toolkit [Young et al., 2006] was used for feature extraction, AM training and speech recognition. The SRI LM toolkit [Stolcke, 2002] was used for LM training and perplexity testing. Details of these components will be described in the following sections.

	Perplexity			
	Meeting	Web(ICSI)	Web(Fisher topics)	Web(Swbd)
Sentence-deletion	106.8	192.5	188.5	214.4
Term-deletion	198.0	399.4	376.5	425.8
Count-deletion	103.3	208.3	190.5	217.8

Table 3.9: The perplexity of word-based 3-gram models trained from 4 subsets of the training corpora with 3 purging approaches. 'Sentence-deletion' means deleting the whole sentence if an OOV term is found; 'Term-deletion' means deleting the OOV terms only and retaining the rest of the sentence; 'Count-deletion' means deleting OOV terms from intermediate statistics. The subset *Meeting* includes ICSI/AMI+fisher+Swbd/CHE which have been given in Table 3.8.

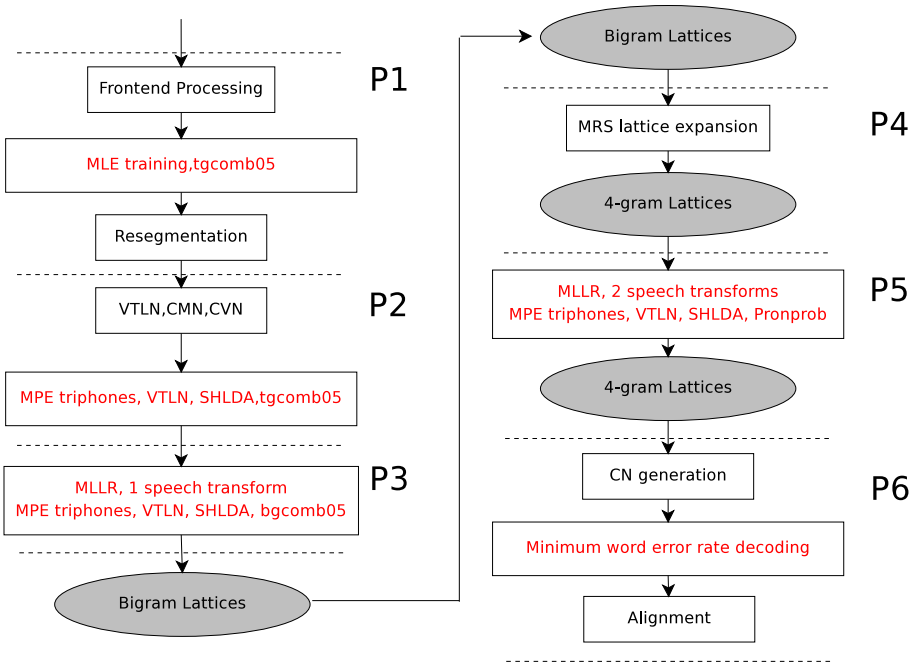

Figure 3.12: The diagram of the AMI RT05s LVCSR system. Reproduced from [Hain et al., 2006a].

Feature extraction

The Mel frequency cepstral coefficients (MFCC) was selected to represent speech. The 12 primary MFCC components were augmented by their first and second -order temporal derivatives as well as zero-order coefficients, leading to a 39-dimension feature vector. Speaker-based cepstral mean normalisation (CMN) and variance normalisation (CVN) were applied to compensate for channel variation.

Model name	Training corpora	# Shared states
STDAM.phoneme.org	icsinistislam05	65812
STDAM.phoneme	icsinistislami05-purged	56642

Table 3.10: The acoustic models trained for the imitative LVCSR system and the LVCSR baseline system. *STDAM.phoneme.org* was trained from the whole speech data and *STDAM.phoneme* was trained from the OOV-purged speech data. The numbers of shared states are reported in the last column to show the model size.

Acoustic model training

We chose triphones as the units to model the acoustic features. The phone set includes 44 non-silence phones plus a long silence /*sil*/ and a short silence /*sp*/. The non-silence phones compose dictionary words, while the long silence represents the leading and ending silence of an utterance, and the short silence represents optional pauses between words. To model context dependency, a phone within the context of its two neighbours is modelled by a *triphone*, and each triphone corresponds to a HMM with 3 non-skip states. Note the short silence model /*sp*/ has only one skippable state.

To prevent data sparsity caused by scarce contexts, decision tree-based clustering was applied to tie states of similar triphones, by asking some phonetic questions that were borrowed from the AMI RT05s system. For each tied state, a Gaussian mixture distribution was used to represent the emission probability. The number of Gaussian components was tuned to optimise the LVCSR performance on the development set.

As mentioned, we built the LVCSR baseline system by first imitating the AMI RT05s system, and then purging the OOV terms from the training materials. Therefore we trained two sets of triphone models: one was trained on the whole speech data, used by the imitative system, and the other was trained on the OOV-purged speech data, used by the OOV-free system. Table 3.10 summarises these two sets of models.

Language model training

We chose classical 3-gram models as the LMs for the LVCSR systems, and applied Kneser-Ney discounting plus interpolation for smoothing. As the training data is large in volume, we first trained small-scale models on each individual corpus, and then merged them into a single model by interpolation.

Again, we had to train two sets of models for the imitative system and the OOV-free

Model name	LM type	LM order	Training corpora	Perplexity
AMILM.word.3g	word	3-gram	-	84.3
STDLM.word.3g.org	word	3-gram	STDTXT	83.8
STDLM.word.3g	word	3-gram	STDTXT-purged	78.4

Table 3.11: The perplexity of three word-based 3-gram LMs for the LVCSR systems. *STDLM.word.3g.org* was trained on the whole text data while *STDLM.word.3g* was trained on the OOV-purged text data. *AMILM.word.3g* is the model used by the AMI RT05s system. Results were computed on the transcript of the RT05s evaluation set.

system respectively, for which training the former used all the text data while training the latter used the OOV-purged data. The perplexities of these two sets of models are reported in Table 3.11. For comparison, the perplexity of the model used by the actual AMI RT05s system is also reported.

The results in Table 3.11 show that our model trained on the whole data performs as well as the AMI RT05s model, with respect to perplexity. The model trained on the OOV-purged data exhibited lower complexity, which can be attributed to the reduced vocabulary size.

Decoding

To transcribe the input speech, time synchronous decoding was conducted using the HTK tool *HDecode*. Decoding parameters, especially the insertion penalty and LM scale factor, were tuned to optimise the recognition performance on the development set, according to the two-phase tuning strategy presented in Section 3.2.3.

3.4.2 Experimental results

Table 3.12 presents the experimental results of the trained LVCSR systems on the evaluation set, in terms of word error rate (WER). The 'official' result of the AMI RT05s P1 system reported by Hain et al. [2006a] is given first, and then the imitative system and the LVCSR baseline system (OOV-free) are presented.

The results in Table 3.12 indicate that our imitative system performed as well as the AMI RT05s P1 system. The minor difference ($\approx 0.2\%$) might arise from the feature extraction (AMI RT05s used features based on perceptual linear prediction (PLP), while our system used MFCCs), segmentation (we got the segmentation from the con-

System	AM	LM	Dictionary	WER%
AMI RT05s	-	-	-	34.9
Imitative LVCSR	STDAM.phoneme.org	STDLM.word.3g.org	AMI05s	35.1
LVCSR Baseline	STDAM.phoneme	STDLM.word.3g	AMI05s-purged	39.5

Table 3.12: The LVCSR performance of our LVCSR systems in terms of WER. The first row cites the performance of the AMI RT05s P1 system [Hain et al., 2006a], and the second and third row gives the performance of our imitative system and our LVCSR baseline system, respectively.

catenation time marked (CTM) reference, while the AMI05 used a manual segmentation) and tuning process (we used *rt04sdev* for system tuning, while AMI RT05s used *rt04seval*).

With OOV terms purged out, as in the case of the LVCSR baseline system, a 4.4% absolute increase in WER was observed, leading to about a 40% WER. This is a typical result for ASR on spontaneous speech, which reveals how challenging it is to transcribe spontaneous speech.

3.5 STD baseline system

According to the STD architecture, an STD system consists of an ASR subsystem and an STD subsystem. For the ASR subsystem, we made use of the models trained for the LVCSR baseline system; for the STD subsystem, we designed the term detector based on the tool *Lattice2Multigram* developed by Brno University of Technology (BUT) [Szöke et al., 2005a]. This section describes the implementation of the STD baseline system, and reports the experimental results.

3.5.1 ASR subsystem

Implementation

The ASR subsystem transcribes input speech to word or phoneme lattices, depending on whether it is a word or phoneme -based system. For the word-based system, acoustic models and language models trained for the LVCSR baseline system were reused directly, whilst for the phoneme-based system, we could only reuse the acoustic models, and train phoneme-based n-gram LMs on phoneme text converted from the

Model name	LM type	LM order	Training corpora	Perplexity
STDLM.phn.2g	phoneme	2-gram	STDTXT-purged	16.2
STDLM.phn.3g	phoneme	3-gram	STDTXT-purged	9.5
STDLM.phn.4g	phoneme	4-gram	STDTXT-purged	6.8
STDLM.phn.5g	phoneme	5-gram	STDTXT-purged	5.7
STDLM.phn.6g	phoneme	6-gram	STDTXT-purged	5.3

Table 3.13: The Perplexity of phoneme-based n-gram LMs trained on OOV-purged text data and used for the ASR subsystem.

OOV-purged training text by the pronunciation dictionary, following the same process used to train the word-based LMs. Perplexities of the phoneme-based LMs are shown in Table 3.13. As expected, higher orders lead to lower perplexity, but the marginal perplexity decrease becomes smaller and smaller as the LM order increases.

Experimental results

With the acoustic and language models ready, we conducted word-based and phoneme-based speech recognition. Besides single-best transcripts, we used the recogniser to generate word or phoneme lattices, depending on the system type. As for the experimental results, we report the WER for word-based systems and phoneme error rate (PER) for phoneme-based systems, plus the lattice density computed as the average number of nodes per second, following the definition of the SRILM tool [Stolcke, 2002].

For the word-based system, we tested the case of using a 3-gram LM; for the phoneme-based system, we tested the cases of using LMs whose orders are from 4 to 6. LMs whose order was lower than 4 gave too big lattices, while LMs whose order was higher than 6 made the recognition too slow. Note that some parameters related to lattice density were tuned before hand to optimise STD performance on the development set, including the number of tokens per state and the pruning beam width, which were then kept unchanged in system development.

Table 3.14 shows the result of the word-based subsystem and Table 3.15 shows the results of the phoneme-based subsystems. As expected, amongst phoneme-based systems, higher order LMs gave better recognition performance, and generated more sparse lattices. This can be ascribed to more strict linguistic constraints introduced by applying higher order of LMs. This argument also explains why the word lattices are

System type	LM order	WER%	Lattice density
word	3-gram	39.5	622

Table 3.14: The performance of the word-based ASR subsystems for lattice generation. The performance, in terms of WER, is based on the best path in the lattice.

System type	LM order	PER%	Lattice density
phoneme	4-gram	44.53	6199
phoneme	5-gram	41.65	1648
phoneme	6-gram	40.49	805

Table 3.15: The performance of the phoneme-based ASR subsystems for lattice generation. The performance, in terms of PER, is based on the best path in the lattice.

more sparse than phoneme lattices. Note that direct comparison of WER/PER of word and phoneme -based systems is meaningless as they are calculated based on different vocabularies.

3.5.2 STD subsystem

Implementation

The STD subsystem searches the lattices generated by the ASR subsystem for query terms. We have discussed in Section 3.1.3 that enquiry terms must be converted to compatible search forms before being processed. For a word-based system, the conversion is trivial, whereas for a phoneme-based system, the conversion might be complex. Although a dictionary look-up is enough for INV terms, LTS models must be used to predict pronunciations for OOV terms.

For the baseline system, we chose the LTS model based on a classification-and-regression-tree (CART). With this model, the pronunciation of a grapheme in a term is determined by examining its left and right grapheme neighbours. We used a CART implementation designed for Festival, a full-fledged speech synthesis system built in the Centre for Speech Technology Research (CSTR), University of Edinburgh [Clark et al., 2007].

In experiments, we followed the CART training described by Black et al. [1998]. We first designed an *allowable table* that specifies possible pronunciations of each

grapheme, using trail and errors, until 95% of the training exemplars were aligned successfully. Afterwards, a tree was trained for each grapheme using the AMI RT05 dictionary with OOV terms purged. We experimented with various configurations, especially the 'stop' value which specifies the minimum number of exemplars required to split a node in the tree. The best performance was obtained with the stop value setting of 1, which accords with the observation reported by Black et al. [1998].

With the lattices and search forms ready, the enquiry terms were searched for by matching the search forms to partial paths in the lattices, as discussed in Section 3.1.3. In the baseline system, we just took the exact match. Lattice-based Baum-Welch confidence was used to measure the reliability of each detection, and the term-independent decision strategy was used to make the hit/FA decision, with the decision threshold estimated by optimising STD performance on the development set.

System naming

A practical concern in describing the experiments and reporting the results is that we will have many results to report for various systems based on various configurations, which often makes the presentation confusing. To give a clear presentation, we assign each system a nick name in the format $(wrd|phn)*.(l|p)(i|t|d)*[.(cart|jmm)*]$, where the first part denotes the lattice unit or system type, and the second part denotes the confidence measurement and decision strategy, and the third part denotes the LTS model, if required. Note that each part can be appended by some additional description to make further clarification. Table 3.16 lists the meanings of the accepted options in the naming format.

As an example, a name 'phn.fi.cart' represents a phoneme-based system that utilises lattice-based confidence and term-independent hit/FA decision, employing a CART-based approach to predict pronunciations for OOV terms.

Experimental results

Following the naming convention, we report the experimental results with the word and phoneme -based STD systems in Table 3.17 and 3.18 for INV terms and OOV terms respectively. The ATWV is the main metric in evaluation, and the max-ATWV reports the best performance with an ideal decision threshold. In addition, the false alarm rate and miss rate corresponding to the ATWV are also reported to give more details of system behaviour.

Option	Specifying	Meaning	Reference section
wrd	system type	word-based system	
phn	system type	phoneme-based system	
l	confidence	lattice-based confidence	
p	confidence	direct posterior confidence	7.1
i	decision strategy	term-independent decision	
t	decision strategy	term-dependent decision	5.1.3
d	decision strategy	discriminative decision	6.2.2
cart	LTS	cart-based approach	
jmm	LTS	joint-multigram model-based approach	5.1.2

Table 3.16: The meaning of the accepted options in the naming format.

System name	System type	LM order	ATWV	max-ATWV	P(FA)	P(Miss)
wrd.fi	word	3-gram	0.5661	0.6191	0.00002	0.410
phn.fi	phoneme	4-gram	0.2438	0.3569	0.00003	0.723
phn.fi	phoneme	5-gram	0.3670	0.4470	0.00004	0.595
phn.fi	phoneme	6-gram	0.4173	0.4988	0.00004	0.542

Table 3.17: The STD performance of the baseline systems on INV terms. 'P(FA)' and 'P(Miss)' denote the false-alarm rate and miss rate respectively.

Figure 3.13 shows the DET curves of the word and phoneme -based STD systems on INV terms, and Figure 3.14 shows the DET curves of the phoneme-based STD systems on OOV terms. Note that the word-based system can not detect OOV terms, therefore does not appear in Figure 3.14.

The results shown above confirm that a word-based system tends to outperform a phoneme-based system on INV terms, which can be attributed to the lexical constraints that are available for word-based systems. We also find that higher order LMs gave better performance for phoneme-based systems. In fact, applying a 6-gram phoneme LM, the phoneme-based system was approaching the word-based system in terms of ATWV.

On OOV terms, it is not surprising that the word-based system detected nothing so got a zero ATWV. For the phoneme-based systems, some occurrences were captured, but the detection accuracy was rather poor. In fact, the high number of false-alarms drove the ATWV down below zero. Higher order LMs again gave better performance.

System name	System type	LM order	ATWV	max-ATWV	P(FA)	P(Miss)
wrd.li	word	3-gram	0.0	0.0	0.0	0.0
phn.li.cart	phoneme	4-gram	-0.1647	0.0016	0.00030	0.861
phn.li.cart	phoneme	5-gram	-0.1232	0.0084	0.00029	0.829
phn.li.cart	phoneme	6-gram	-0.1010	0.0088	0.00028	0.816

Table 3.18: The STD performance of the baseline systems on OOV terms. 'P(FA)' and 'P(Miss)' denote the false-alarm rate and miss rate respectively.

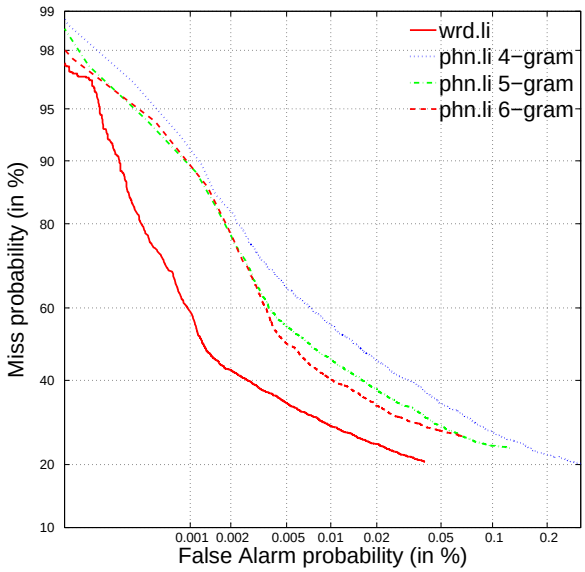

Figure 3.13: The DET curves of word and phoneme -based systems on INV terms. 'wrd.li' is the word-based system using a 3-gram LM, and 'phn.li' is the phoneme-based system working with various LMs.

In the rest of this thesis, the phoneme-based system applying a 6-gram LM is taken as the *phoneme-based STD baseline system*, which is *phn.li* for INV terms and *phn.li.cart* for OOV terms. The word-based system *wrd.li* applying a 3-gram LM is correspondingly taken as the *word-based STD baseline system*.

3.5.3 Comparing to the NIST evaluation

It is difficult to compare the performance of different STD systems because the experimental conditions are usually substantially different. However, it is still useful to look at some results from contemporary research, so that we can have a rough idea whether

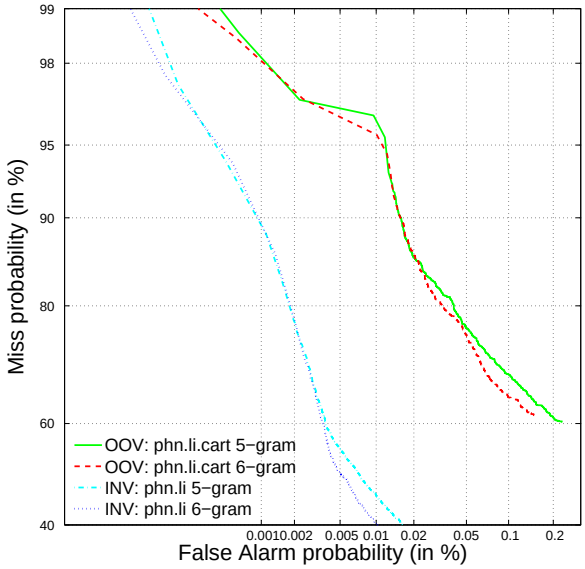

Figure 3.14: The DET curves of phoneme-based systems on OOV terms. For comparison, results on INV terms are shown as well.

the experimental results we obtained are convincing. For that purpose, we consider the NIST 2006 STD evaluation to be a good reference.

First of all, we notice that the experimental conditions in our work are significantly different from those in the NIST evaluation: (1) our experiments were conducted on individual head microphone (IHM) speech, while the NIST evaluation was conducted on multiple distance microphone (MDM) speech, which is more noisy; (2) the NIST evaluation was not particularly concerned with OOV terms, and hence the OOV rate was relatively low. Akbacak et al. [2008] estimated that the OOV rate was 0.03% with a 60K vocabulary in the BNEWS task, and 0.18% with a 10k vocabulary.

Referring to Table 1.2, we can see that the best result in terms of ATWV reported in the NIST evaluation under the meeting condition is 0.2553, which was achieved by SRI [Fiscus et al., 2006]. This number is lower than our results on INV terms but higher than those on OOV terms. Considering the different experimental conditions, we conclude the results we obtained with the baseline systems are reasonable.

3.6 Summary

We have presented the background of our experiments, including the experimental framework and configurations, the data profile, and the implementation of the baseline

systems. Comparing with the results from the NIST 2006 STD evaluation, we are confident that the results we obtained so far are good baselines.

Examining these results, we can see that the baseline system is very weak on OOV terms. Thorough inspection shows that the poor performance arises from the high number of false alarms, and that many false alarms are due to incorrect pronunciations that were predicted by the CART-based LTS model. This suggests that we should look for more suitable LTS models, for example, a model with fewer independence assumptions. In the next chapter, we will investigate such a model based on joint multigrams, which will give higher quality of pronunciation predictions.

Chapter 4

Joint-multigram model-based pronunciation prediction

Examining the detection results of the baseline system, we found that a large portion of the detection errors arose from the incorrect pronunciations predicted by the letter-to-sound (LTS) module, which is based on CARTs in the baseline system. A supposed problem of the CART model is that it assumes the pronunciation of a word is completely determined by the spelling form, and the phonemes in the pronunciation are conditionally independent given the spelling. This assumption is obviously not true since phonetic rules do apply and regulate pronunciation. Another disadvantage of the CART model is that it is not suitable for predicting multiple pronunciations and it is hard to assign a suitable confidence measure to the predicted pronunciation. To overcome these shortcomings, we studied the joint-multigram model, which allows general dependence among graphemes and phonemes, and can predict multiple pronunciations.

In this chapter, we first introduce the motivation and formalisation of the joint-multigram model, and compare it with other models such as CARTs and HMMs. Afterwards, we present our extended work on training and prediction, and report the experimental results.

4.1 Joint-multigram model

4.1.1 Motivation

Conditional model

In general, a LTS model can be regarded as a stochastic mapping between two streams of symbols, spelling G and pronunciation Q, where $G = (g_1, g_2, ..., g_L)$, is a grapheme sequence, and $Q = (q_1, q_2, ..., q_R)$, is a phoneme sequence. The best pronunciation $\hat{Q}$ of a word with spelling G can be defined as the Q that possesses the maximum posterior probability $P(Q|G)$, written as

$$\hat{Q} = \arg\max_{Q} P(Q|G). \tag{4.1}$$

We can factor $P(Q|G)$ to elementary probabilities so that Equation 4.1 can be computed in practice. Various independence assumptions can be applied, giving rise to various LTS models. For example, if phonemes q_i are assumed to be conditionally independent given G, and phoneme q_i is totally determined by a window of graphemes $\tilde{g}_i$ that is centred on the grapheme corresponding to q_i, we get the CART model, giving the posterior probability in Equation 4.1 as

$$P(Q|G) = \prod_i P(q_i|\tilde{g}_i). \tag{4.2}$$

Generally speaking, if a LTS model factors $P(Q|G)$ into elementary probabilities that are conditioned on graphemes only, it is called a *conditional model*. The underlying idea of conditional models is that a pronunciation Q is a subsequent random process derived from spelling G, and is determined by G exclusively. A decision tree is a typical conditional model which stores $P(q_i|\tilde{g}_i)$ in the tree nodes; a neural network trained with windowed graphemes as input and corresponding phonemes as output is another conditional model, representing a non-linear mapping from $\tilde{g}_i$ to $P(q_i|\tilde{g}_i)$.

Relationship of writing and speaking

Conditional models hold the assumption that the written form determines the spoken form. However, this assumption is not really true, although it accords with our psychological experience. We note that there are many homophones and polyphones, and many pronunciations break pronunciation rules. This means that the relationship between spelling and pronunciation is rather complex and that spelling can not fully

determine pronunciation. In fact, writing and speaking are two systems developed from the same social background, therefore closely related; on the other hand, these two systems developed for different purposes, so are independent and follow different principles.

In order to account for the special relationship between writing and speaking, a mutual-dependence view was proposed by Deligne et al. [1995]. According to this perspective, people always hold some intention before writing and speaking, and this intention gives rise to both spelling and pronunciation. Following this idea, human language can be regarded as a *hidden process*, from which derive the observable processes corresponding to spelling and pronunciation, and each of which follows respective principles. This is illustrated in Figure 4.1.

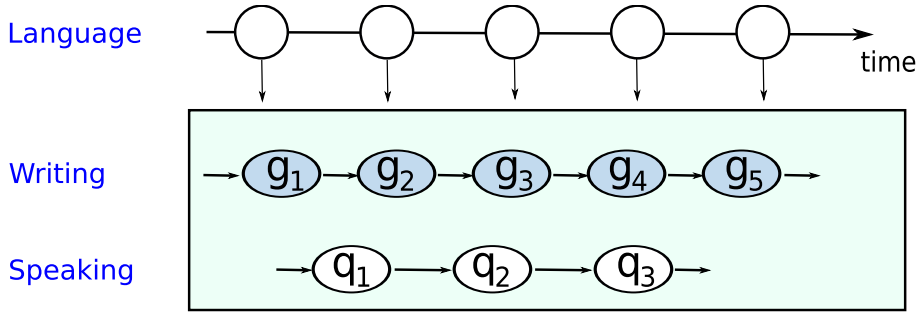

Figure 4.1: The hidden process of human language which gives rise to writing and speaking.

This view of human language motivates a *joint model*, which describes the joint probability of spelling and pronunciation. To derive a formal representation, we start by investigating the correspondence between phonemes and graphemes.

Grapheme-phoneme correspondence

A *GP correspondence* is defined as an allowable alignment between the spelling and the pronunciation of a word; accordingly, a pair of grapheme and phoneme sequences derived from the correspondence is defined as a GP pair. The simplest correspondence assumes that one phoneme corresponds to one grapheme, and if the spelling and pronunciation are of different length, null symbols are inserted [Black et al., 1998]. More complex correspondences are possible. As an example, Figure 4.2 shows a correspondence of the word ‘THOUGHT’, where the mapping between phonemes and graphemes is many-to-many. Note that other correspondences may be valid. For example, ‘GH’ can be attached either to the preceding ‘O’ to correspond with ‘[ao]’ or

attached to the following 'T' to correspond with '[t]'.

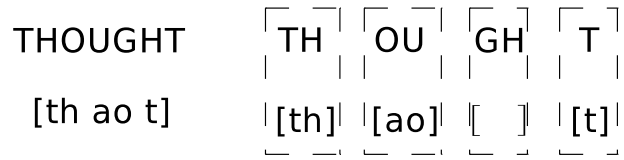

Figure 4.2: An example of GP correspondence of the word 'THOUGHT'.

This example shows that a GP correspondence should be a many-to-many mapping, and should be stochastic. A multigram-based stochastic model is designed to account for these properties.

Joint-multigram and graphone

A multigram is a symbol sequence whose length could be 0,1, or more. A GP pair contains a grapheme multigram and a phoneme multigram, and so is also called a *joint-multigram*. Following Bisani and Ney [2002], we call a joint-multigram $u = \{\tilde{g}, \tilde{q}\}$ a *graphone*, where $\tilde{g}$ is the grapheme component and $\tilde{q}$ is the phoneme component. Using this definition, a GP correspondence is formulated as a process U that is a sequence of graphones, and thus we have

$$U = (u_1, u_2, ..., u_K) \tag{4.3}$$

$$= \begin{pmatrix} \tilde{g}_1, \tilde{g}_2, ..., \tilde{g}_K \\ \tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_K \end{pmatrix} \tag{4.4}$$

where K is the number of graphones in the correspondence, and $\tilde{g}_i$ and $\tilde{q}_i$ satisfy the following constraint

$$\begin{pmatrix} \tilde{g}_1 \frown \tilde{g}_2 \frown ... \frown \tilde{g}_K \\ \tilde{q}_1 \frown \tilde{q}_2 \frown ... \frown \tilde{q}_K \end{pmatrix} = \begin{pmatrix} g_1 g_2 g_3 ... g_L \\ q_1 q_2 q_3 ... q_R \end{pmatrix} \tag{4.5}$$

with the symbol $\frown$ denoting concatenation, L and R being the length of the grapheme and phoneme sequences respectively.

Note that both $\tilde{g}_i$ and $\tilde{q}_i$ contain variable lengths of symbols. Hence the many-to-many mapping of a GP correspondence has been addressed by the graphone representation.

Joint-multigram model

To describe the stochastic property of the GP correspondence, we can model the probability distribution over process U, leading to the *joint-multigram model* or *graphone model*, formally written as:

$$P(U) = P(u_1, u_2, ..., u_K). \tag{4.6}$$

This model was initially proposed by Deligne et al. [1995], and has been applied to LTS [Bisani and Ney, 2002; Galescu and Allen, 2002; Chen, 2003; Vozila et al., 2003], LVCSR [Bisani and Ney, 2003a, 2005] and STD [Akbacak et al., 2008]. With the joint-multigram model, the joint probability of spelling G and pronunciation Q is calculated as a probability summation over all possible graphone sequences. Thus we obtain

$$\begin{aligned} P(G,Q) &= \sum_{U;G(U)=G,Q(U)=Q} P(U) && (4.7) \\ &= \sum_{U;G(U)=G,Q(U)=Q} P(u_1, u_2, ..., u_K) && (4.8) \end{aligned}$$

where $G(U)$ and $Q(U)$ are the grapheme and phoneme sequences corresponding to process U. Then the task of pronunciation prediction formulated in Equation 4.1 can be cast as an optimisation with respect to U, expressed by Equation 4.9-4.10.

$$\begin{aligned} \hat{Q}(G) &= \arg\max_{Q} P(G,Q) && (4.9) \\ &= \arg\max_{Q} \sum_{U;G(U)=G,Q(U)=Q} P(U) && (4.10) \end{aligned}$$

4.1.2 Formulating training and prediction

Maximum-likelihood (ML) Training

Suppose we have a set of words and their pronunciations, denoted as $\Phi = \{\phi_j\}$, where $\phi_j = (G_j, Q_j)$ is an *exemplar*, representing the j-th word and its pronunciation. Applying Equation 4.7, and assuming all the words are independent, the total log probability of Φ is written as

$$\Gamma_{\Phi}(\theta) = \sum_{j=1}^{N} logP(\phi_j;\theta) \tag{4.11}$$

$$= \sum_{j=1}^{N} logP(G_j,Q_j;\theta) \tag{4.12}$$

$$= \sum_{j=1}^{N} log(\sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_i} P(U_j;\theta)) \tag{4.13}$$

where U_j is a possible correspondence for exemplar ϕ_j, and θ denotes the parameters of the joint-multigram model $P(U)$. $\Gamma_{\Phi}(\theta)$ in Equation 4.11 is a likelihood function of the parameters θ of $P(U)$. By maximising $\Gamma_{\Phi}(\theta)$ with respect to θ, we obtain a model trained in the sense of maximum-likelihood (ML), formally written as

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{N} log(\sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_i} P(U_j;\theta)) \tag{4.14}$$

where $\hat{\theta}$ denotes the optimal parameters.

To get a computational algorithm, we need to formulate the generic form $P(U)$ into a product of elementary probabilities, by applying some independence assumptions. Deligne et al. [1995] assumed that all graphones are mutually independent, leading to a unigram graphone model, given by

$$P(U_j;\theta) = \prod_{i=1}^{|U_j|} P(u_{ji};\theta) \tag{4.15}$$

where u_{ji} is the i-th graphone of the correspondence U_j, and $|U_j|$ is the number of graphones in U_j. Note that the independence assumption among graphones is quite different from the independence assumption among phonemes in the CART model, because the phonemes within a graphone are obviously dependent.

Extending the unigram graphone model to n-gram models is straightforward, which generally improves the modelling power as wider contexts are concerned [Bisani and Ney, 2002; Chen, 2003]. Equation 4.16 shows the factorisation of $P(U)$ under the n-gram graphone model,

$$P(U_j;\theta) = \prod_{i=1}^{|U_j|} P(u_{ji}|h_{ji};\theta) \tag{4.16}$$

where h_{ji} is the graphone history of u_{ji}.

Because the unigram model is just a special case of the n-gram model, we will use the n-gram form to derive the training algorithm. For that, we substitute for $P(U_j)$ in Equation 4.14, giving rise to

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=j}^{N} log(\sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_j} \prod_{i=1}^{|U_j|} P(u_{ji}|h_{ji};\theta)). \quad (4.17)$$

Note that optimising θ equals to optimising the n-gram model $P(u|h)$, and therefore Equation 4.17 can be written as a more useful form,

$$P(u|h;\hat{\theta}) = \arg\max_{P(u|h;\theta)} \sum_{j=1}^{N} log(\sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_j} \prod_{i=1}^{|U_j|} P(u_{ji}|h_{ji};\theta)). \quad (4.18)$$

There is no closed-form solution for Equation 4.18. An EM algorithm, devised by Deligne et al. [1995], has become a standard approach to tackle the optimisation problem. By the EM algorithm, model parameters are re-estimated iteratively until the value of the likelihood function converges to a local maximum. The re-estimation formula is given by

$$P(u|h;\theta') = \frac{\sum_{i=j}^{N} \sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_i} P(U_j;\theta) n_{u,h}(U_j)}{\sum_{u'} \sum_{j=1}^{N} \sum_{U_j;G(U_j)=G_j,Q(U_j)=Q_j} P(U_j;\theta) n_{u',h}(U_j)} \quad (4.19)$$

where $P(u|h;\theta')$ is the updated model and $n_{u,h}(U)$ is the number of occurrences of graphone u with history h in the graphone sequence U.

A simplified form of Equation 4.19 is given by substituting the probability of the most probable correspondence for the summarised probability over all possible correspondences. This leads to a *Viterbi* re-estimation, shown in Equation 4.20,

$$P(u|h;\theta') = \frac{\sum_{j=1}^{N} n_{u,h}(U_j^*)}{\sum_{u'} \sum_{j=1}^{N} n_{u',h}(U_j^*)} \quad (4.20)$$

where U_j^* represents the correspondence of ϕ_j with the highest probability among all possible correspondences.

Exact and approximated prediction

With the joint-multigram model, the task of pronunciation prediction is casted to the inference of the best pronunciation of a word given its spelling. The inference process is also called *decoding*.

The decoding program can be implemented according to Equation 4.10 by applying the n-gram model, giving the exact prediction as below:

$$\hat{Q}(G) = \arg\max_{Q} \sum_{U;G(U)=G} P(U;\theta) \tag{4.21}$$

$$= \arg\max_{Q} \sum_{U;G(U)=G} \prod_{i=1}^{|U|} P(u_i|h_i;\theta) \tag{4.22}$$

where $U = \{u_i\}$ is an arbitrary possible correspondence.

A critical problem of the exact prediction is that the search space is very huge. To reduce the computation, the probability of the most probable correspondence is again used to approximate the summarised probability of all correspondences, leading to the *Viterbi prediction*, given by Equation 4.23. It has been shown that this approximation does not jeopardise performance much by Wang et al. [2009a] and Bisani and Ney [2008].

$$\hat{Q}(G) = Q(\arg\max_{U;G(U)=G} P(U)) \tag{4.23}$$

$$= Q(\arg\max_{U;G(U)=G} \prod_{i=1}^{|U|} P(u_i|h_i;\theta) \tag{4.24}$$

An interesting property of the joint-multigram model is that graphemes and phonemes are symmetric in the model, so this model can be applied to predict the spelling from a pronunciation as well, e.g., [Galescu and Allen, 2002]. Deriving the formula for this inverse prediction is straightforward and similar to that we did for the spelling-pronunciation prediction.

4.1.3 Comparing joint-multigram model with CART and HMM

The joint-multigram model has exhibited better performance than other statistical models on the LTS task, such as CARTs and HMMs, e.g., [Chen, 2003; Bisani and Ney, 2008]. In this section, we will study various LTS models in a general graphical representation so that we can reach a deeper understanding of how the joint-multigram gains superiority over other models and what the potential disadvantages are.

Following the idea that spelling and pronunciation are derived from the same underlying process, the probabilistic structure described by a joint-multigram model can be illustrated as the dependence among graphones, as represented by the graphical model

in Figure 4.3. Figure 4.4 gives another representation which explicitly describes the dependence between graphemes and phonemes. For comparison, the graphical models of CARTs and HMMs are presented in Figure 4.5 and 4.6 respectively.

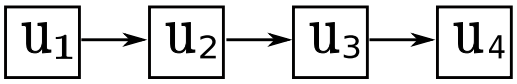

Figure 4.3: The graphical representation of a joint-multigram model where a 2-gram model is assumed.

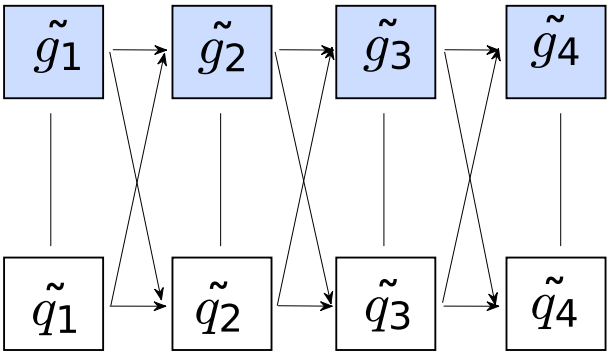

Figure 4.4: The graphical representation of the joint-multigram model, explicitly representing the dependence between graphemes and phonemes. The shaded nodes denote observable variables, and the unshaded nodes denote hidden variables.

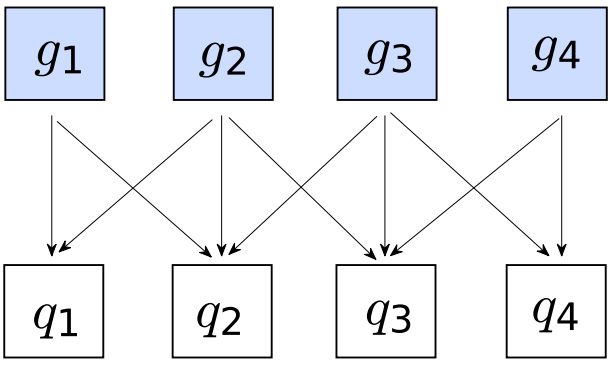

Figure 4.5: The graphical representation of a CART model. The shaded nodes denote observable variables, and the unshaded nodes denote hidden variables.

Inspecting the probabilistic structures of these three models, we find that both the CART and HMM make unrealistic independence assumptions. In CARTs, phonemes

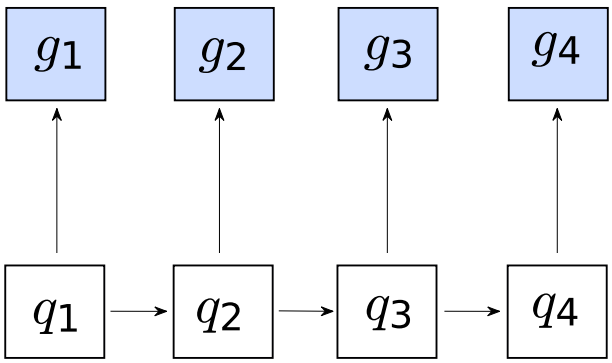

Figure 4.6: The graphical representation of a HMM. The shaded nodes denote observable variables, and the unshaded nodes denote hidden variables.

are assumed to be mutually independent and in HMMs, graphemes are assumed to be conditionally independent given phonemes. In reality, however, we know that phonotactic and morphological rules regulate spelling and pronunciation, and hence introduce dependence among phonemes and graphemes. These unrealistic assumptions inevitably affect model accuracy. A distinct advantage of the joint-multigram model is that it allows flexible dependence among phonemes and graphemes.

Modelling the flexible dependence leads to a multitude of advantages for the joint-multigram model: **(1)** By removing unrealistic assumptions, it can model the obvious dependence among graphemes, phonemes and graphones, leading to a more precise model; **(2)** Modelling these dependence gives a globally-optimised LTS conversion. This is especially important for n-best prediction; **(3)** The graphones derived in model training reveal some morphemic rules that are useful for applications such as automatic inventory acquisition.

On the other hand, the joint-multigram model also suffers some problems: **(1)** The complex dependency leads to complex models, raising the risk of data sparsity and over-fitting, which degrades generalisation performance; **(2)** The universal dependency makes it hard to extract explicit LTS rules; **(3)** The left-dependent assumption in the n-gram model misses the right context[1] that is assumed to be important.

In summary, because of the ability to describe flexible probabilistic structure, the joint-multigram tends to be superior to other models on the LTS task, at least when training data are abundant.

[1]This problem does not come from the joint-multigram model itself, but from the limited graphone history within the n-gram model. If the order of the n-gram model is infinite, the left-dependence assumption and right-dependence assumption are equal in modelling power.

4.2 Implementation

In this section, we present our implementation for the joint-multigram model, including data preparation, model training and prediction. The focus will be put on the novel techniques we devised for improving the prediction accuracy.

4.2.1 Data preparation

Our experiments were conducted on the AMI RT05 dictionary which contains 50740 words. Normalisation was firstly applied to clear the dictionary, including: (1) eliminating word fragments, e.g., 'AUSE'; (2) eliminating acronyms, e..g, 'U.S.A', 'IEEE'; (3) eliminating digits, e.g., '20522'; (4) eliminating non-alphabet symbols, e.g., '$([-'.

After the normalisation, 8000 words were randomly selected from the dictionary for evaluation, leaving 36575 words for training and 4064 words for development. Each word plus its pronunciation compose an exemplar, forming the training set, development set and evaluation set for the experiment.

4.2.2 Model training

We adopted the Viterbi approach to train the joint-multigram model, according to Equation 4.20. For that, we devised a 4-step training strategy, as shown in Figure 4.7, to improve training efficiency. Specifically, we first built a primary bi-gram model from scratch, and then applied this primary model to segment the training exemplars into graphone sequences via a forced alignment, from which higher order n-gram models are trained. The major benefit of this *segmentation & n-gram training* approach is that we can employ an existing LM toolkit to build various orders of n-gram models[2].

Model initialisation The initial graphone unigram model was built by collecting all potential graphones (GP pairs). The occurrence $c(u_i)$ for each graphone u_i was counted by a trellis match and then was normalised into probabilities. The initialisation process is described by the pseudo program in Figure 4.8.

[2]We tried to iterate the training as well, i.e., apply the trained n-gram model to re-segment the training exemplars and then update the n-gram model again. However, we did not find any performance improvement. This indicates that the bi-gram model trained according to Equation 4.20 is good enough for segmentation, and the 4-step training strategy does not jeopardise the quality of the resulting models.

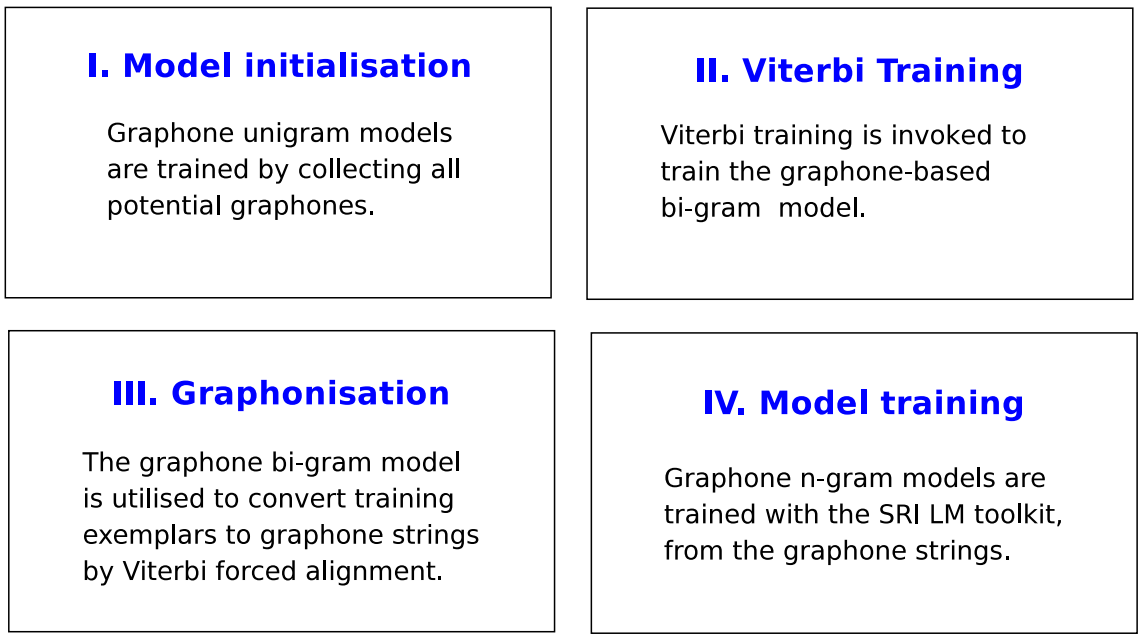

Figure 4.7: The 4-step training strategy for the joint-multigram model. *Graphonisation* means converting the training exemplars to graphone sequences.

Viterbi training With the initial model, the Viterbi training algorithm formulated by Equation 4.20 was invoked to train a bi-gram model. In this process, each training exemplar was first segmented into the best graphone string via a Viterbi forced alignment that will be described shortly, and then a graphone bi-gram model was trained from the resulted graphone strings. The segmentation and model training were iteratively conducted until the likelihood of the training data became stable.

Graphonisation *Graphonisation* means converting the training exemplars to graphone text. This was realised as a forced alignment between the grapheme and phoneme sequence of an exemplar. Figure 4.9 illustrates the alignment process. Note that each segment on the aligned path corresponds to a graphone, so the alignment segments training exemplars into graphone strings. In practice, a dynamic program was designed to perform the forced alignment.

Model training With the graphone text generated by graphonisation, the SRI LM toolkit [Stolcke, 2002] was employed to train n-gram graphone models. Various model configurations and smoothing techniques were examined, as will be reported in Section 4.3.3 and Section 4.3.4.

```
L1: minimum length of a graphone's  grapheme or
     phoneme component
L2: maximum length of a graphone's grapheme or
     phoneme component
u(g,p): a graphone with grapheme g and phoneme p
e(gr,ph): a training exemplar
e.gr{i,j}: grapheme sequence of e between position i and j
e.ph{i,j}: phoneme sequence of e between position i and j

foreach e  in dictionary D
  for (i=0; i<len(e.gr); i++)
    for (j=0; j<len(e.ph);j++)
      for(l=L1; l<=L2; l++)
        for(r=L1;r<=L2;r++)
          if (u(e.gr{i-l+1,i}, e.ph{j-r+1,j}) valid)
              c(u)++; cc++;
          end
        end
      end
    end
  end
end

foreach u
    p(u)=c(u)/cc;
end
```

Figure 4.8: The pseudo program for initialising the joint-multigram model.

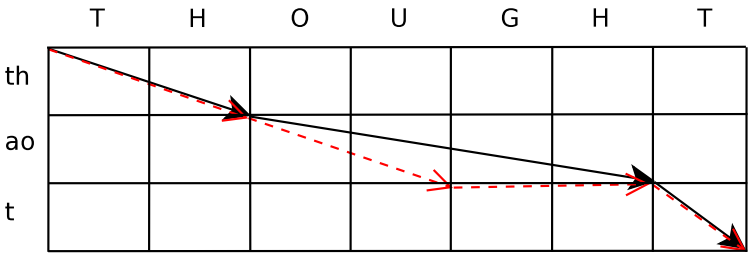

Figure 4.9: Forced alignment and graphonisation of the word 'THOUGHT'. Each path shows a possible grapheme-phoneme alignment, and each segment of the path corresponds to a graphone. For example, the solid path represents a graphone sequence $(u(TH,th),u(OUGH,ao),u(T,t))$, and the dot-line represents a graphone sequence $(u(TH,th),u(OU,ao),u(GH,-),u(T,t))$, where '$-$' denotes a null symbol.

4.2.3 Prediction

The prediction was implemented as a Viterbi decoding formulated by Equation 4.23. The pseudo program shown in Figure 4.10 outlines the decoding process. Furthermore, in order to improve the prediction accuracy, we made a couple of extensions, presented as follows.

```
gr: grapheme sequence of the word in question
L1: minimum length of grapheme or phoneme seqeunce
in graphones
L2: maximum length of grapheme or phoneme seqeunce
in graphones
gr{i,j}: subsequence from gr between position i and j

L=length(gr);
for (i=0; i<L; i++)
 for (j=L1; j<=L2; j++)
  foreach u that (u.gr==gr{i-j+1,i}
            && L1<=length(u.ph)<=L2)
     u.score=max(u'.score+LM(u',u))
           where u' in (vec[i-j]);
     u.hist=u' that (u'.score+LM(u',u)==u.score);
     push u into vec[i];
  end
 end
end

ub=best u in vec[L-1];
trace-back(ub);
```

Figure 4.10: The dynamic program used to predict pronunciations from word spellings with the joint-multigram model.

Insertion compensation The first concern arises from comparing predictions of different length, or with different numbers of phonemes in the predicted pronunciations. The basic algorithm assumes that the probability of short and long predictions are comparable, but this is dubious, because a longer prediction usually has a lower probability, leading to a disadvantage when competing with shorter predictions. Borrowing the idea of insertion penalties from speech recognition, we introduced an *insertion*

compensation to compensate long predictions. Letting ς denote the insertion compensation, the decoding algorithm of Equation 4.23 is updated as follows,

$$\hat{Q}(G) = Q(\arg \max_{U;G(U)=G} P(U) + \varsigma |Q(U)|) \tag{4.25}$$

where $|\cdot|$ denotes the length of a symbol sequence.

Forward-backward decoding We have mentioned that a potential problem of the joint-multigram is that it models the left context of a graphone but loses its right context, even though the right context may be more important [Hallahan, 1995]. To amend this weakness, we studied the backward decoding, i.e., conducting the decoding from right to left. Furthermore, we combined the forward and backward decoding to give additional performance improvement.

N-best prediction The accuracy of pronunciation prediction, even with the joint-multigram model, is still imperfect. A possible solution to the error-prone prediction comes from so called *n-best decoding*, which provides multiple predictions so that correct pronunciations are more likely to be covered.

Based on the joint-multigram model, we implemented n-best prediction by keeping n alternative graphones when extending partial paths, which generates a graphone lattice when the decoding is completed. With the graphone lattice, n-best predictions are obtained by selecting the best n paths whose final nodes get the highest probability. The confidence of each prediction is computed as the ratio of the probability of the graphone path of this prediction and the probability accumulation of the whole lattice, as expressed as follows,

$$c(U) = \frac{P(U)}{\sum_{U' \subseteq \Re(\phi)} P(U')} \tag{4.26}$$

where $\Re(\phi)$ denotes the graphone lattice generated in decoding for word ϕ, and $P(U)$ denotes the probability of an arbitrary graphone path U in $\Re(\phi)$. The accumulated probability of the lattice is often called the *evidence* of ϕ, formally written as Equation 4.27.

$$E(\phi) = \sum_{U' \subseteq \Re(\phi)} P(U') \tag{4.27}$$

Note that Bisani and Ney [2008] presented the same idea of n-best prediction. Our work is independent and contemporary, and we further applied the n-best prediction to STD, as proposed in the next chapter.

4.3 Experimental results

We report the experimental results in this section. The data sets are those presented in Section 4.2.1, and the programs for the joint-multigram model training and decoding were implemented in C. We first describe the metrics for LTS performance evaluation, and then report the results with the CART-based approach and the joint-multigram model-based approach respectively.

4.3.1 Evaluation metrics

The accuracy of pronunciation prediction can be measured by two metrics: word error rate (WER) and phoneme error rate (PER). The WER is computed as the proportion of incorrect predictions, written as

$$WER = 1 - \frac{N_{correct}}{N_{total}} \tag{4.28}$$

where N_{total} denotes the number of words whose pronunciations are predicted, and $N_{correct}$ is the number of words whose pronunciations are correctly predicted. Herein a prediction is correct only if the prediction matches the canonical pronunciation exactly. If a word has multiple pronunciations (which is the case of the AMI dictionary), the prediction is assumed to be correct if it matches any of these pronunciations.

Phoneme errors are computed as the edit distance between phoneme strings of the predicted pronunciation and the canonical pronunciation, and the PER is defined as the proportion of the phone errors, formally written as

$$PER = \frac{\sum_i (L_i^{ins} + L_i^{del} + L_i^{sub})}{\sum_i L_i} \tag{4.29}$$

where L_i denotes the length of the canonical pronunciation of word i, and L_i^{ins}, L_i^{del}, L_i^{sub} are insertion, deletion and substitution errors of the predicted pronunciation. Again, if a word has multiple pronunciations, the best matched pronunciation is assumed in the calculation and other pronunciations are ignored.

4.3.2 CART-based approach

We chose the CART-based approach [Black et al., 1998] implemented in the Festival system [Clark et al., 2007] as a reference. As a state-of-the-art speech synthesis system, Festival has been thoroughly designed and widely used, so it is meaningful to use it as the reference point.

In brief, a CART is a decision tree that determines the pronunciation of a letter given its grapheme context. In such a tree, a node maintains a list of phonemes that the letter can be pronounced, and is associated with a binary question regarding the letter's grapheme context. Figure 4.11 gives such a tree for example.

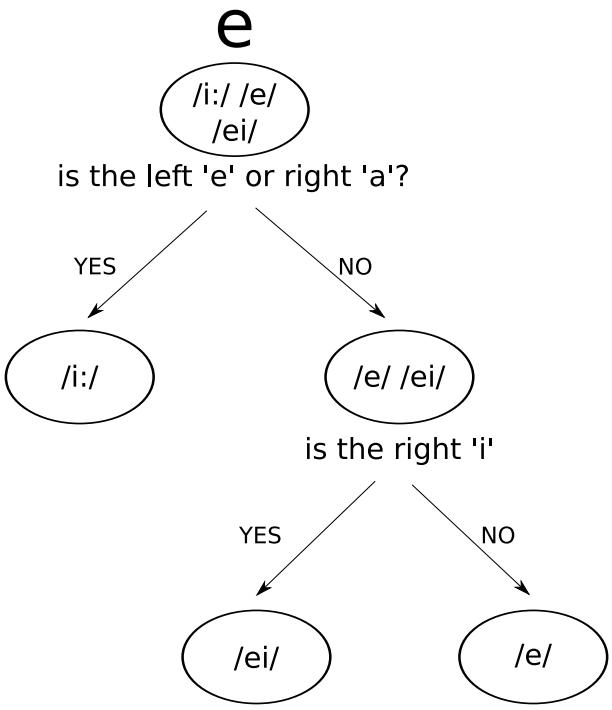

Figure 4.11: An exemplary CART for pronunciation prediction. This tree predicts pronunciations for the letter 'e'.

In the training phase, words and their pronunciations in the training set are first aligned, and then a CART corresponding to a particular letter is trained with all the grapheme-phoneme correspondences whose grapheme component is this letter. A minimum entropy criteria is applied to direct the node splitting and tree growing. In prediction, the pronunciation of a letter within a particular grapheme context is obtained by traversing the CART that corresponds to this letter, starting from the root node and stopping at a leaf node. In each step, the question associated to the current node of the tree is answered by looking at the grapheme context of the letter, according to which

the next node to traverse is determined. This procedure is iterated until a leaf node is arrived, at which the most likely pronunciation assigned to this leaf is taken as the pronunciation of the letter. Once the pronunciation of a single letter can be predicted, the pronunciation of a word can be predicted by concatenating the pronunciations of all letters of the word.

In implementation, we followed the training process presented by Black et al. [1998]. First of all, we built an *allowable table* that specifies the allowable pronunciations of a letter. This was a trial & error process, until 95% of the training exemplars were aligned successfully. Afterwards, a CART was built for each letter, with cross-validation on the development set. We experimented with various configurations, especially the 'stop' value which specifies the minimum number of training exemplars that should be distributed to a node when growing the tree. The best performance was achieved when setting the stop value to 1, meaning that a leaf node may have just one training exemplar. This observation is consistent to the the results reported by Black et al. [1998].

The experimental results are shown in Table 4.1. The WER and PER are reported for both the training and evaluation set. These results are comparable to the English results reported by Black et al. [1998], although we used a different dictionary.

	Evaluation set		Training set	
STOP	WER%	PER%	WER%	PER%
1	**35.2**	**8.7**	14.0	3.4
2	39.1	9.6	23.9	5.5
3	40.0	9.8	27.7	6.4

Table 4.1: The performance of the CART-based LTS system on both the evaluation and the training set. 'STOP' represents the minimum number of training exemplars that should be distributed to a leaf node when growing the trees.

4.3.3 Graphone size and model order

The first set of experiments we conducted with the joint-multigram model were designed to find optimal settings for the graphone size and the order of the graphone model. The graphone size is defined as the minimum and maximum length of the grapheme and phoneme components of a graphone, denoted as *NN*. The graphone

model order is the order of the n-gram model, denoted as M. The graphone size and model order together determine the model complexity, and therefore should be considered in conjunction.

Various combination of NN and M were experimented with on the evaluation set, whose results are shown in Figure 4.12. This figure shows that for each NN, the optimal model order M is a medium value. This is understandable because if the order is too low, the model is less powerful, and if it is too high, data sparsity will set in. Another observation is that the prediction performance was more impacted by the graphone size rather than the model order, which may be attributed to the fact that the graphone size determines the graphone inventory, which affects the model power more radically than the strength of the probabilistic dependency among graphones that is determined by the model order.

The best performance on the development was achieved when setting $NN = 1-2$ and $M = 4$; this configuration would be applied in the following experiments, although $M = 3$ seems slightly better on the evaluation set according to Figure 4.12.

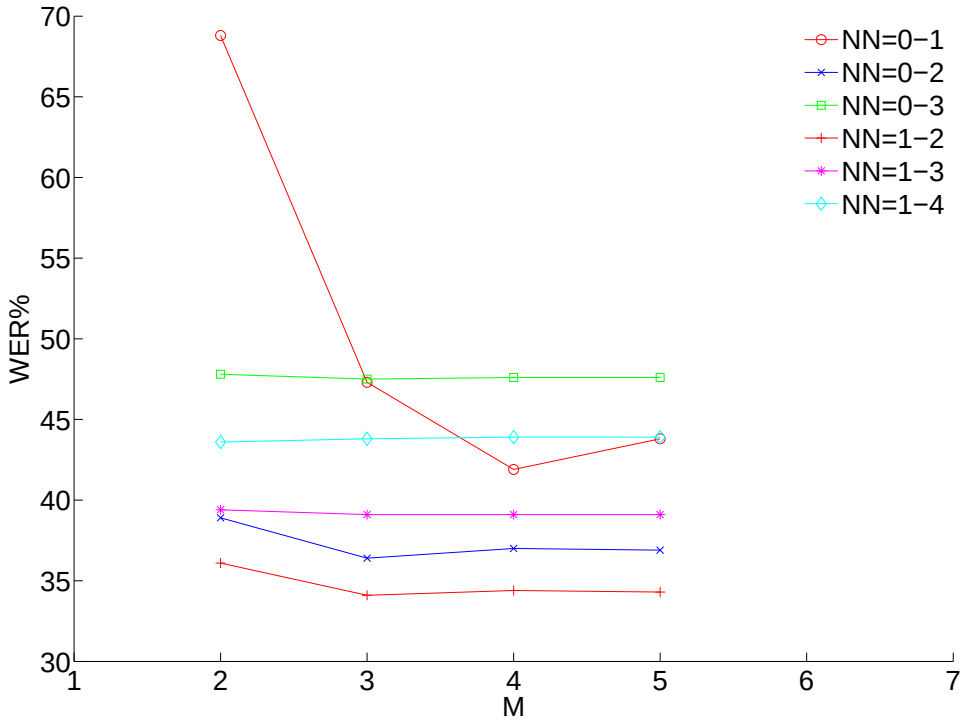

Figure 4.12: The performance of the joint-multigram model-based pronunciation prediction with various settings of graphone size NN and model order M, in terms of WER.

Table 4.2 presents the detailed results with the selected configuration in the second row. We observe that the joint-multigram model, even without any enhancement discussed shortly, outperformed the CART.

	Evaluation set		Training set	
Model	WER%	PER%	WER%	PER%
CART (stop=1)	35.2	8.7	14.0	3.4
joint-multigram model	34.4	8.5	13.3	3.0
+ Kneser-Ney discounting & interpolation [Sec. 4.3.4]	33.2	8.2	12.4	2.9
+ insertion compensation [Sec. 4.3.5]	32.7	8.1	12.5	2.9
+ backward decoding [Sec. 4.3.6]	31.3	7.8	10.4	2.4
+ 50-best decoding [Sec. 4.3.7]	30.9	7.7	10.0	2.3
+ pronunciation unification [Sec. 4.3.8]	30.3	7.5	9.7	2.3

Table 4.2: The performance of the joint-multigram model-based pronunciation prediction, on both the training and evaluation set.

4.3.4 Model smoothing

We mentioned that the joint-multigram model describes a rather complex stochastic structure and hence is susceptible to data sparsity and over-fitting. Therefore a suitable smoothing is important. The SRI LM toolkit implemented various smoothing techniques. To find out which technique is the best for smoothing the joint-multigram model, we experimented with most of the smoothing options provided by the SRI LM toolkit. The results on the development set are reported in Table 4.3.

The results in Table 4.3 indicate that all the smoothing techniques improved the prediction accuracy, among which the Kneser-Ney discounting plus interpolation gave the best results. Applying this method, we conducted the experiments on the evaluation set, and obtained the results shown in the third row of Table 4.2.

4.3.5 Insertion compensation

This experiment tested the contribution of the insertion compensation as presented in Section 4.2.3. We first optimise the prediction performance on the development set with respect to the compensation value. Figure 4.13 shows the tuning results, from which we found that 1.3 was a good choice for the compensation. This value was then

Smoothing algorithm	WER%	PER%
no smoothing	33.0	8.2
absolute discounting	32.0	8.0
absolute discounting+interpolation	32.0	8.0
natural discounting	32.5	8.1
natural discounting+interpolation	32.5	8.1
Witten-Bell discounting	32.4	8.1
Witten-Bell discounting+interpolation	32.0	8.0
Kneser-Ney discounting	32.9	8.2
Kneser-Ney discounting+interpolation	**32.0**	7.8
modified Kneser-Ney discounting	33.4	8.4
modified Kneser-Ney discounting+interpolation	32.2	7.9

Table 4.3: The performance of various smoothing techniques applied to the joint-multigram model. Results are reported on the development set. The best result is in bold face.

applied to conduct experiments on the evaluation set, giving the results shown in the fourth row of Table 4.2. The results confirm that using the insertion compensation does improve the prediction accuracy.

4.3.6 Forward and backward decoding

This experiment tested the backward decoding. For comparison, we used the same graphone text for the forward and backward decoding, except that the graphone sequences were reversed. The results from the backward decoding, reported in the fifth row of Table 4.2, are better than the results from the forward decoding, confirming that modelling right contexts indeed improves the prediction performance.

Furthermore, we tried to combine the predictions made by the forward and the backward decoding. Specifically, we checked the pronunciations predicted by these two methods, and selected the pronunciation with either higher probability or higher confidence. Experimental results are presented in Table 4.4, which show that the confidence-based combination gave more additional improvement.

Figure 4.13: The performance of the joint-multigram model-based LTS system on the development set, with various insertion compensation.

	Evaluation set		Training set	
Model	WER%	PER%	WER%	PER%
forward decoding	32.7	8.1	12.5	2.9
backward decoding	31.3	7.8	10.4	2.4
dictionary combination (probability)	31.3	7.7	10.4	2.4
dictionary combination (confidence)	**31.1**	7.7	10.3	2.4

Table 4.4: The performance of the forward and backward decoding applied to the joint-multigram model. 'Probability' and 'confidence' denote the probability-based combination and the confidence-based combination respectively. The best result is shown in bold face.

4.3.7 N-best decoding

We tested the n-best decoding with the joint-multigram model in this experiment. We consider that a n-best prediction is correct if any of the predictions in the n-best is correct. To examine the n-best performance, we generated 50 predicted pronunciations

for each word, and sorted them in order of confidence. The WER of the n-best prediction with various n is shown in Figure 4.14 for both the forward and the backward decoding. It is interesting to see that when n is relatively small, the backward decoding outperforms the forward decoding; with n increased, the decoding direction becomes unimportant.

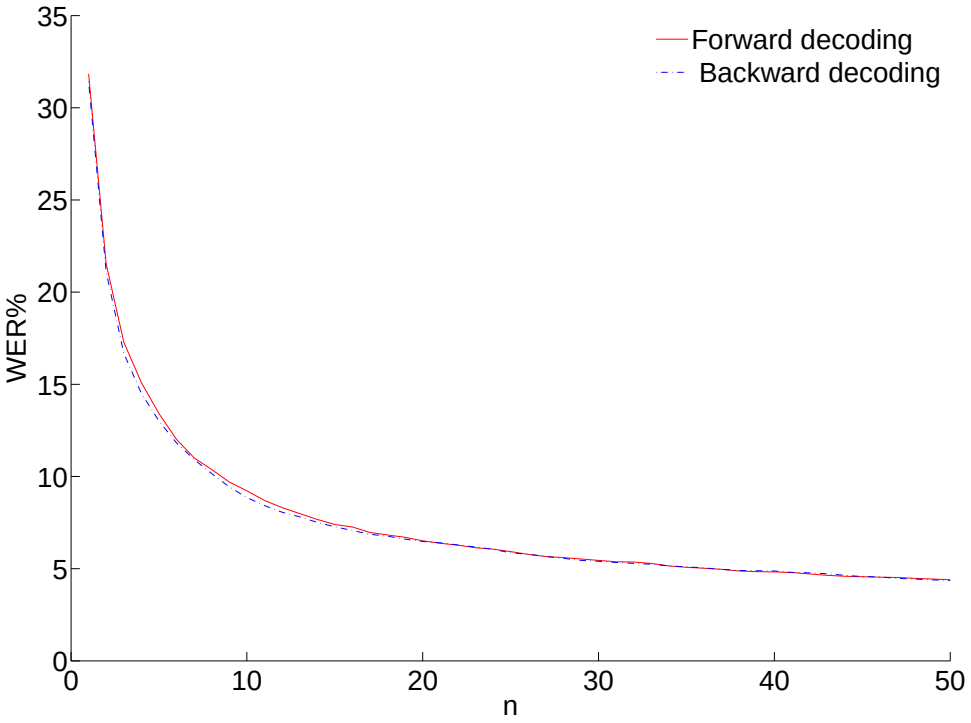

Figure 4.14: The performance of the n-best prediction with the joint-multigram model, with n ranging from 1 to 50.

Note that the n-best decoding not only predicts multiple predictions, but also impacts the best prediction in our implementation. This is because multiple partial paths with different phoneme histories are retained in the n-best decoding, whereas in the 1-best decoding, we just retain the 1-best partial paths. Therefore the best prediction with n-best decoding tends to be more accurate than the best prediction with 1-best decoding. The results of the best prediction using 50-best decoding are shown in Table 4.5. An unexpected observation is that the combination did not improve the performance, indicating that the forward and backward decoding do not add complementary information with the n-best decoding.

	Evaluation set		Training set	
Model	WER%	PER%	WER%	PER%
forward decoding	31.3	7.8	10.7	2.5
backward decoding	**30.9**	7.7	10.0	2.3
dictionary combination (confidence)	31.2	7.7	11.0	2.6

Table 4.5: The performance of the best prediction with 50-best decoding. Both the forward and backward decoding, as well as their confidence-based combination, are presented. The best result is shown in bold face.

4.3.8 Pronunciation unification

All the experiments reported so far are based on the Viterbi decoding formulated by Equation 4.23, which is an approximation to the exact decoding formulated by Equation 4.21. We have mentioned that this approximation is necessary because the searching space of the exact decoding is too large to implement the decoding in practice.

The graphone lattice generated from the n-best decoding provides an opportunity for exact decoding. Constrained by the paths of a lattice, the exact decoding can be conducted in a subset of the searching space, and so becomes tractable[3]. In implementation, we merged predictions in the n-best list that correspond to the same pronunciation, and then re-ordered the n-best list according to the merged confidence. We call this approach *pronunciation unification*. Note that pronunciation unification is not equal to exact decoding because the lattice is only a subset of the whole search space; nevertheless, it does approach the exact decoding, in a tractable way. The experimental results are reported in Table 4.6, which confirm that unification does provide some performance improvement, though not substantial.

4.3.9 Complementarity with CART

We postulate that the joint-multigram model and the CART are complementary, due to their very different probabilistic structures. The joint-multigram model tends to describe long-span dependency among phonemes and graphemes, while the CART tends to look at more broader context; the joint-multigram is good at finding general pronunciation "rules" while the CART is capable of remembering special cases. To test this supposed complementarity, we tested an *Oracle combination* of these two

[3]The original idea of the lattice-based exact decoding came from Bisani and Ney [2008].

	Evaluation set		Training set	
Model	WER%	PER%	WER%	PER%
forward decoding	30.7	7.6	10.4	2.4
backward decoding	**30.3**	7.5	9.7	2.3

Table 4.6: The prediction performance with the pronunciation unification. The best 50 predictions were produced and predictions with the same pronunciations are merged into one prediction whose confidence is the sum of all the merged predictions. The prediction list is then re-ordered according to the merged confidence. The WER and PER of the best prediction after unification are reported.

models. With this combination, we examine the predictions made by the CART and the joint-multigram model, and treat a word being correctly predicted if either of these two models predicts it correctly. This means that the results of the Oracle combination reflect the best performance that we could achieve if we combine the predictions from these two models in an ideal way. Table 4.7 shows the experimental results.

	Evaluation set		Training set	
Model	WER%	PER%	WER%	PER%
CART (stop=1)	35.2	8.7	14.0	3.4
joint-multigram	30.3	7.5	9.7	2.3
Oracle(CART+joint-multigram)	19.1	4.2	3.0	0.6

Table 4.7: The best performance of the CART and joint-multigram model -based LTS approaches, as well as their Oracle combination.

We see that the results of the Oracle combination are much better than the results with each individual model, which suggests that if we had an ideal way to combine the predictions of these two models, the prediction accuracy would be significantly improved. In addition, we observe that the performance of the 2-best prediction with the joint-multigram model is similar to the performance of the oracle combination of the joint-multigram model and CART, which indicates that the joint-multigram model-based n-best prediction produces the correct pronunciation in even a short n-best list, as can be seen in Figure 4.14.

4.4 Summary

In this chapter, we studied joint-multigram model-based pronunciation prediction. We introduced the motivation and the formulation for the joint-multigram model, and presented our implementation. Extended work on insertion compensation, backward decoding and n-best prediction were proposed. Experimental results show that the joint-multigram model clearly outperformed the CART model on the task of pronunciation prediction.

The goal of studying the LTS models is to provide reliable pronunciations for OOV terms in STD. In the next chapter, we will apply the joint-multigram model to predict pronunciations for OOV terms, which will give the first improvement to our STD baseline system.

Chapter 5

Stochastic pronunciation modelling

We have presented the joint-multigram model for the LTS task in the previous chapter; in this chapter, we apply this model to generate pronunciations for OOV terms for the STD task. A salient advantage of the joint-multigram model is that multiple pronunciations can be easily obtained with n-best decoding. Taking account of these multiple pronunciations may recover some errors of the 1-best prediction and hence give higher detection accuracy.

The multiple predictions, plus the associated confidence measures, reflect the uncertainty of the pronunciations of OOV terms. This uncertainty does not come from ASR errors, but rather the stochastic relationship between spelling and pronunciation. This insight leads to a stochastic pronunciation model (SPM), defined as a probability distribution over the predicted pronunciation. With the SPM, the task of detecting an OOV term is cast as the task of detecting any of its possible pronunciations, giving rise to a Bayesian treatment of OOV terms.

In the following sections, we first apply the 1-best pronunciation predicted by the joint-multigram model to the STD task, and then extend to n-best pronunciations. The SPM will be proposed afterwards, and compared to and combined with a soft match-based approach.

5.1 Joint-multigram model-based pronunciation prediction for STD

5.1.1 Predicting pronunciations for OOV terms

We have demonstrated that the joint-multigram model outperforms the CART baseline when used to predict pronunciations for *held-out* words. Now we apply this model to predict pronunciations for OOV terms for STD. In order to make this comparison, we trained a joint-multigram model and a CART model on the same data, then predict pronunciations for the OOV terms. The training data include all the words in the AMI dictionary except the OOV terms, which is 48250 words in total. We first examine the accuracy of the predicted pronunciations, as reported in Table 5.1. Note that some OOV terms do not exist in the AMI dictionary, so we have only 382 terms for evaluation.

Model	WER%	PER%	Hit	Substitution	Insertion	Deletion
CART	30.1	8.5	2006	102	10	73
joint-multigram	31.7	8.5	2017	136	22	28

Table 5.1: The results of pronunciation prediction based on the CART and the joint-multigram model. The evaluation set contains 382 terms that appear in our OOV term list but have reference pronunciations in the original AMI dictionary (before OOV purge). 'Hit', 'Substitution' and 'Deletion' are three types of errors when computing the phoneme error rate.

We can make some interesting observations from Table 5.1: (1) the joint-multigram model-based approach did not achieve better performance than the CART-based approach on OOV terms, which is different from the results we achieved on held-out words; (2) the patterns of the prediction errors based on these two models are different: the CART-based approach produced more deletions while the joint-multigram model-based approach produced more insertions, and the joint-multigram model-based approach obtained more hits at the cost of more substitutions.

The discrepancy between the results on held-out words and OOV terms with these two models is not surprising. We have discussed in the previous chapter that the joint-multigram model is good at learning rules but weak on remembering special cases, thus it has a disadvantage when predicting pronunciations for irregular OOV terms. How-

ever, it is not fair to conclude that the joint-multigram model is inferior to the CART, because the test set is very small, and the reference pronunciations are not reliable. Moreover, the word error rate of the predicted pronunciations is not an indicator of the wholeness of these pronunciations for STD, as will be seen in the following section.

5.1.2 Application to spoken term detection

We now apply the pronunciations generated by the joint-multigram model to detect the OOV terms. We followed the same detection procedure as in the baseline system, but substitute the pronunciations generated by the CART model for those generated by the joint-multigram model. Table 5.2 reports the experimental results.

System name	Model	ATWV	max-ATWV	P(FA)	P(Miss)
phn.li.cart	CART	-0.1010	0.0088	0.00028	0.816
phn.li.jmm	joint-multigram	0.0229	0.0273	0.00016	0.819

Table 5.2: The STD performance on OOV terms with pronunciations predicted by the CART and the joint-multigram model. *phn.li.cart* is the phoneme-based STD baseline system which uses the CART model, and *phn.li.jmm* is a similar system except that the pronunciations were predicted by the joint-multigram model.

We observe that the pronunciations predicted by the joint-multigram model led to a better STD performance in terms of ATWV, although they contain more prediction errors in terms of WER when compared to the pronunciations predicted by the CART model. We also find that the joint-multigram model-based prediction gave a lower false alarm rate, although at the cost of a higher miss rate.

5.1.3 Confidence normalisation

An observation from Table 5.2 is that the ATWV is worse than the max-ATWV in both systems, which suggests that the confidence threshold tuned on the development set and used by the decision maker is not optimal for the hit/FA decision. Further analysis shows that in the CART-based system, a large portion of correct detections were rejected by an over rigorous threshold, and in the joint-multigram model-based system, a number of incorrect detections were accepted by an over loose threshold. These failures suggest that a term-independent threshold is not suitable for an STD

system, especially for OOV terms, because they exhibit a wide range of values for the confidence measures, which caused the global threshold to fail.

To solve this problem, we developed a term-dependent threshold. The idea came from the ATWV-oriented threshold proposed by Miller et al. [2007] and adopted by some other researchers (e.g., Vergyri et al. [2007]; Parlak and Saraçlar [2008]). Here we derive this technique from an alternative perspective: confidence normalisation.

Letting $d = (K, s, v_a, v_l, ...)$ represent a detection of term K, we start from the ATWV definition discussed in Section 3.1.4. For convenience, it is has reproduced here,

$$ATWV = 1 - \frac{\sum_{K \in \Delta} [P_{miss}(K) + \beta P_{FA}(K)]}{|\Delta|}. \tag{5.1}$$

In this definition, the miss rate P_{miss} and false alarm rate P_{FA} are defined as follows,

$$P_{miss}(K) = 1 - \frac{N^K_{hit}}{N^K_{true}} \tag{5.2}$$

$$P_{FA}(K) = \frac{N^K_{FA}}{N^K_{NT}}. \tag{5.3}$$

where N^K_{hit}, N^K_{FA} and N^K_{true} respectively represent the number of hits, false alarms and true occurrences of term K. N^K_{NT} denotes the number of no-target terms, which can be estimated as

$$N^K_{NT} = T - N^K_{true} \tag{5.4}$$

where T is the length of the searched audio in seconds. By simple arrangement, we have

$$ATWV = \frac{1}{|\Delta|}\left(\sum_K \frac{N^K_{hit}}{N^K_{true}} - \beta \frac{N^K_{FA}}{T - N^K_{true}}\right) \tag{5.5}$$

This definition indicates that if a putative detection is a hit, it will provide benefit $\frac{1}{N^K_{true}}$, and if it is a false alarm, it will introduce a cost $\frac{\beta}{T - N^K_{true}}$. Therefore the expected benefit of a putative detection d can be estimated as,

$$\zeta(d) = \frac{c(d)}{N^K_{true}} - \beta \frac{1 - c(d)}{T - N^K_{true}} \tag{5.6}$$

where $c(d)$ is the confidence of d. Considering that any putative detection with positive expected benefit tends to increase the final ATWV, we get the ATWV-oriented decision

strategy:

$$assert(d) = \begin{cases} 1 & if\ \zeta(d) >= 0 \\ 0 & if\ \zeta(d) < 0 \end{cases} \tag{5.7}$$

Note that N^K_{true} is unknown in practice, and thus needs to be estimated from data. Choose the effective occurrence to estimate N^K_{true}, we have:

$$N^K_{true} \approx \sum_i c(d^K_i) \tag{5.8}$$

where d^K_i is the i-th detection of K.

In Equation 5.6, the expected benefit $\zeta(d)$ can be interpreted as a normalisation function ζ_K on $c(d)$, formally expressed as,

$$\zeta_K(c(d)) = \zeta(d). \tag{5.9}$$

We define ζ_K in Equation 5.9 as a *confidence normalisation*. Obviously, ζ_K is term-dependent, and the decision strategy of Equation 5.7 is accordingly a term-dependent decision.

Now we apply the normalisation to the widely used lattice-based confidence defined by Equation 3.18 which is reproduced here for convenience:

$$\begin{aligned} c_{lattice}(d) &= P(K^{t_2}_{t_1}|O) && (5.10) \\ &= \frac{\sum_{C_K} p(O|C_K, K^{t_2}_{t_1}) P(C_K, K^{t_2}_{t_1})}{\sum_\xi p(O|\xi) P(\xi)} && (5.11) \end{aligned}$$

where $K^{t_2}_{t_1}$ denotes the event that K occurs between t_1 and t_2 of the input speech O, C_K is the context of K, and ξ is any path in the lattice. The normalised lattice-based confidence is given by,

$$\zeta_K(c_{lattice}(d)) = \frac{c_{lattice}(d) \times \alpha + \gamma}{\sum_i c_{lattice}(d^K_i)} - \beta \frac{1 - c_{lattice}(d) \times \alpha - \gamma}{T - \sum_i c_{lattice}(d^K_i)} \tag{5.12}$$

where we have introduced a linear transform of $c_{lattice}(d)$ with two adaptable parameters α and γ, to compensate bias of scaling and shift [Siu and Gish, 1999].

From the above derivation procedure, we can see that the confidence normalisation has two aspects of implication: first, it is motivated by maximising the expected contribution to ATWV, and therefore is an ATWV-oriented confidence mapping; second, it is a normalisation for confidence of different terms with different occurrence rates, so it is a term-dependent confidence scaling and shift. The first aspect leads to performance

improvement in terms of ATWV, while the second aspect enhances the entire system in general, especially for detecting OOV terms which are highly diverse.

We conducted the experiments with the confidence normalised, and got the results on INV terms as shown in Table 5.3 and on OOV terms as shown in Table 5.4. In the OOV case, both the CART and joint-multigram model -based systems are reported.

System name	System type	CN	ATWV	max-ATWV	P(FA)	P(Miss)
wrd.li	WORD	NO	0.5661	0.6191	0.00002	0.410
wrd.lt	WORD	YES	**0.5678**	0.5973	0.00002	0.408
phn.li	PHONEME	NO	0.4173	0.4988	0.00004	0.542
phn.lt	PHONEME	YES	0.4743	0.5058	0.00006	0.470

Table 5.3: The STD performance of the word and phoneme -based systems on INV terms. The column *CN* specifies if the confidence normalisation is applied. The best result is shown in bold face.

System name	Model	CN	ATWV	max-ATWV	P(FA)	P(Miss)
phn.li.cart	CART	NO	-0.1010	0.0088	0.00028	0.816
phn.lt.car	CART	YES	0.2126	0.2607	0.00002	0.766
phn.li.jmm	joint-multigram	NO	0.0273	0.0299	0.00016	0.819
phn.lt.jmm	joint-multigram	YES	**0.2761**	0.2770	0.00006	0.667

Table 5.4: The STD performance on OOV terms with pronunciations predicted by the CART and joint-multigram model. The column *CN* specifies if the confidence normalisation is applied. The best result shown in bold face arose from the joint-multigram model-based system with confidence normalisation. Comparing the two systems with confidence normalisation, the joint-multigram model-based system outperformed the CART-based system significantly ($p < 0.001$).

It is interesting to see from Table 5.3 that the confidence normalisation helped both the word and phoneme -based systems, though the improvement to the phoneme system is much more remarkable. Comparing the results on INV and OOV terms, we find that the normalisation gave more substantial performance improvement for the OOV terms. These results support our hypothesis that OOV terms are more idiosyncratic thus more demanding in terms of the confidence normalisation.

To understand how the normalisation improves the decision quality, we examine the discriminative power of the confidence before and after normalisation. For that, we plot the histogram of the confidence of detections, showing hits as green and false alarms as red. The plots are shown in Figure 5.1 and Figure 5.2 for INV terms and OOV terms respectively. In both figures, the two plots on the top show the histograms of hits and false alarms, and the other two on the bottom show the smoothed distribution. Note that the 'irregular' confidence distribution after normalisation may be due to the *term-dependent* scaling and shifting, as well as the numerical precision lost when converting confidence from the logarithm domain to normal probabilities.

We can see that before normalisation, on both the INV and OOV terms, there is a large overlap between hits and false alarms, which indicates that the confidence measure is not powerful enough to discriminate correct and incorrect detections. After normalisation, however, the overlap is substantially shrunk, indicating that more discrimination has been obtained, which then gives rise to higher quality of the hit/FA decision and better STD performance. Because of its universal effectiveness, confidence normalisation will be applied to all the following experiments.

5.1.4 System combination

In Section 4.3.9, we have shown that the CART and the joint-multigram model are complementary. In this section, we consider applying this complementarity to improve the STD performance.

Two approaches were tried: dictionary combination, in which pronunciations predicted by the two models are merged to a single dictionary which is then used to conduct STD; and detection combination, in which STD is conducted separately by two systems, one based on the CART and the other based on the joint-multigram model, and then the detections from both systems are merged in a post-processing step.

The dictionary combination is straightforward, while the detection combination needs some explanation. Basically, we follow an idea similar to the spirit of the ROVER approach [Fiscus, 1997], and rely on the general rule that combining results from complementary systems should improve the system performance. As illustrated in Figure 5.3, detections from each system are aligned and examined in order of time. If a detection does not overlap with a detection from the other system, it is simply copied to the final result along with its confidence. If the same term is hypothesised by both systems at the same time, an output detection is generated which has the earliest

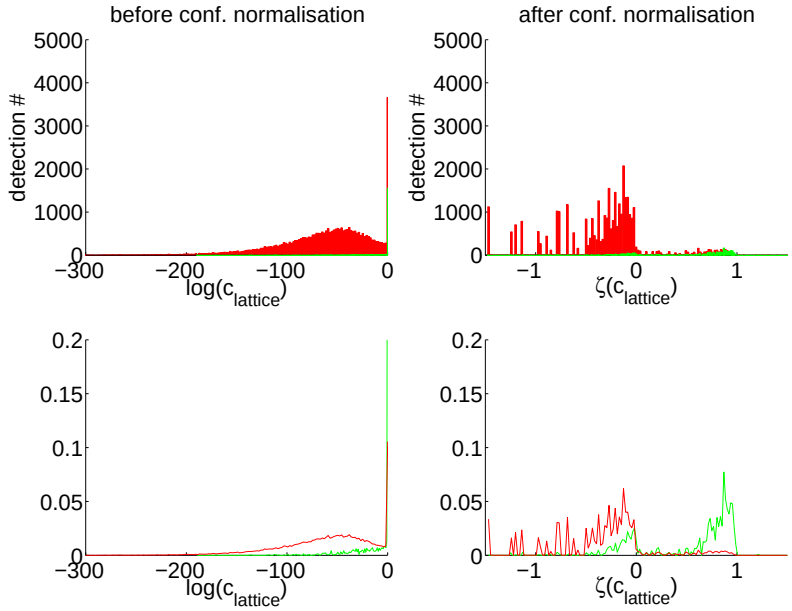

Figure 5.1: The effectiveness of confidence normalisation. The two plots on the top show the distribution histograms and the bottom plots show the smoothed class-conditional distribution. The left two plots show the original lattice-based confidence, and the right two plots show the confidence after normalisation. In each plot, green bars or lines denote hits, and red bars or lines denote false alarms. The experiments were conducted on INV terms.

and latest hypothesised start and end times and a merged confidence according to the following formula:

$$c = 1 - (1 - c_1)(1 - c_2)^{\alpha} \tag{5.13}$$

where c_1 and c_2 are confidence of two individual systems, and α is a fusion factor used to adjust the contribution of individual systems. In experiments, α was tuned to optimise STD performance of the combined system on the development set. Note that this equation applies to non-overlapped detections as well, just treating the confidence of the missed detection as zero and the fusion factor α as 1.

Table 5.5 reports the performance of the systems based on the two combination approaches, and Figure 5.4 shows the DET curves. We find that the two combination approaches both improved the STD performance substantially. A pairwise t-test shows that the improvement from either approach is statistically significant ($p \approx 0.005$), while the detection combination gives better performance when the false alarm rate is low.

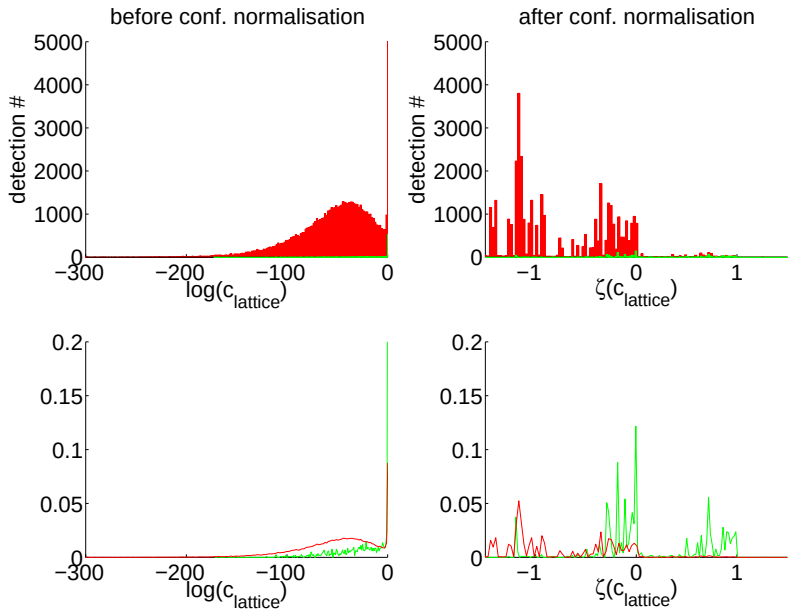

Figure 5.2: The effectiveness of confidence normalisation. The two plots on the top show the distribution histograms and the bottom plots show the smoothed class-conditional distribution. The left two plots show the original lattice-based confidence, and the right two plots show the confidence after normalisation. In each plot, green bars or lines denote hits, and red bars or lines denote false alarms. The experiments were conducted on OOV terms.

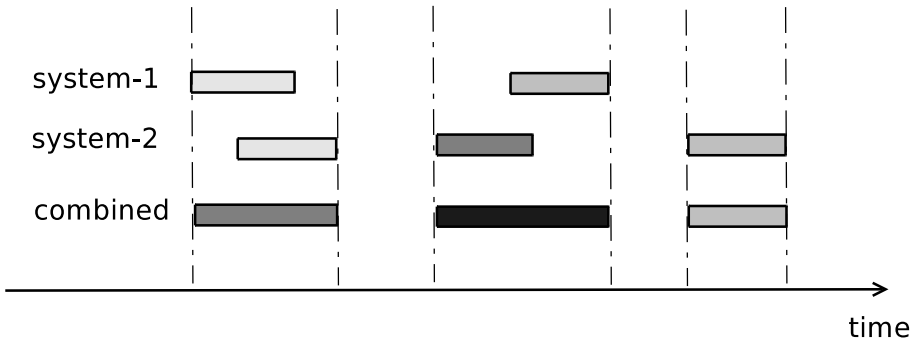

Figure 5.3: Illustration of the process of detection combination. Detections from the two systems that overlap in time are merged as a single detection, while detections without overlap are duplicated in the merged result directly. Confidence is derived according to Equation 5.13 for overlapped detections, and unchanged for non-overlapped detections. Shading represents confidence, with darker being greater.

System name	Combination method	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.cart		0.2130	0.2607	0.00002	0.766
phn.lt.jmm		0.2761	0.2770	0.00006	0.667
phn.lt.cart+jmm.dct	dictionary combination	0.2998	0.3044	0.00007	0.628
phn.lt.cart+jmm.mlf	detection combination	**0.3030**	0.3085	0.00006	0.653

Table 5.5: The STD performance on OOV terms with dictionary combination and detection combination. The best result is shown in bold face. The tuning results showed that $\alpha = 1.0$ gave the best performance for the detection combination, indicating that the two individual systems should be treated equally important.

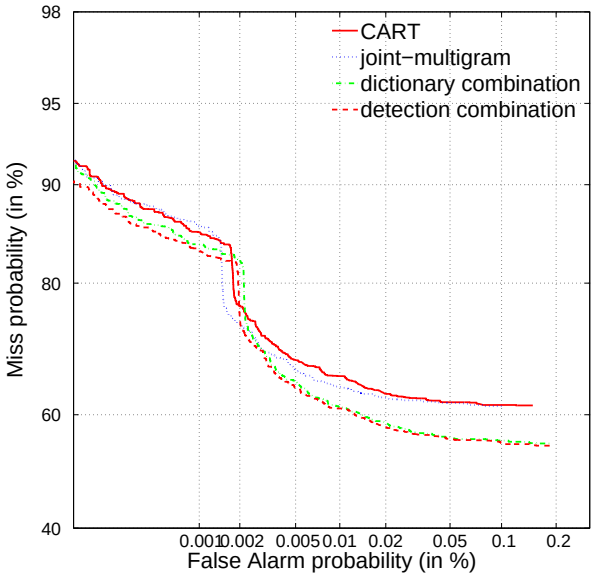

Figure 5.4: The DET curves of the STD systems with pronunciations predicted by the CART and the joint-multigram model. Two combined systems, one based on dictionary combination and the other based on detection combination, are also reported.

5.2 STD with multiple pronunciation prediction

A salient advantage of the joint-multigram model is that it can easily predict multiple pronunciations. With a conditional model, e.g., CART, the term pronunciation is obtained by concatenating the pronunciation of each grapheme that is individually predicted by looking at the local context. By contrast, the joint-multigram model predicts term pronunciations by looking at the probabilities of whole lattice paths, which ensures the n-best predictions are globally optimal.

5.2.1 N-best prediction for OOV terms

With an n-best decoding, a maximum of n pronunciations are generated for a term K, which we denote as Q_1, Q_2, ...Q_n. We further assign a confidence to each pronunciation, denoted as $P(Q_1|G^K)$, $P(Q_2|G^K)$, ... $P(Q_n|G^K)$ where G^K represents the spelling form of K. To control the maximum number of pronunciations delivered to the term detector, a confidence threshold η is set such that any pronunciation Q_i must satisfy the following constraint, otherwise it will be discarded:

$$log(P(Q_i|G^K)) \geq log(P(Q_1|G^K)) - \eta \tag{5.14}$$

where $P(Q_1|G^K)$ is the confidence of the most likely pronunciation.

Figure 5.5 shows the STD performance on OOV terms with n-best predictions. The result shows that in any cases, multiple pronunciations for OOV terms improve the STD performance.

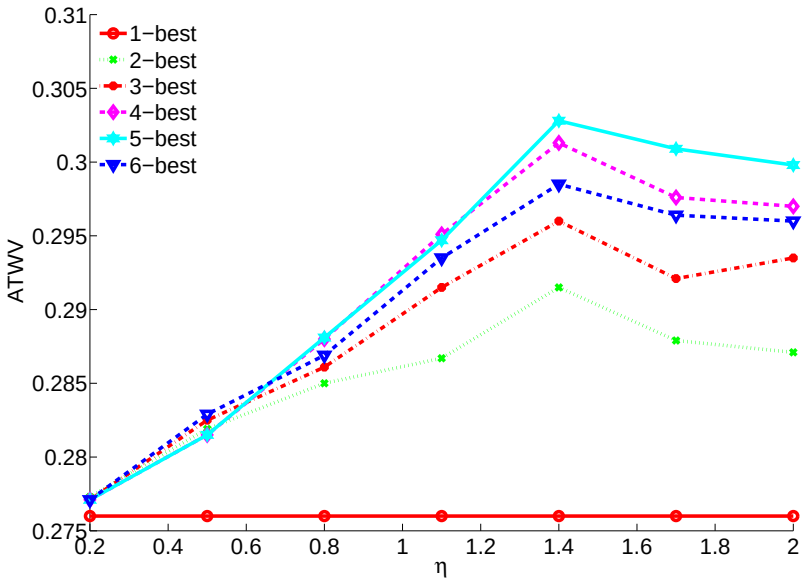

Figure 5.5: The STD performance on OOV terms with n-best predictions. Each curve represents the performance with a particular n and various values of the confidence threshold η.

In practice, we need determine the optimal n and η. This was achieved by tuning these parameters to maximise the STD performance on the development set. The tuning results indicated that n=4 and η=1.4 are optimal. Applying these values, the experiments were carried out on the evaluation set, giving the results shown in Table

5.6, and the DET curves are shown in Figure 5.6. We can see from the results that the n-best prediction substantially improved the STD performance. A pairwise t-test showed that any system using the n-best ($n > 1$) prediction outperformed the 1-best system significantly ($p < 0.01$).

System name	Prediction	#Pron.	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.jmm	1-best	484	0.2760	0.2770	0.00006	0.667
phn.lt.jmm.nbest	n-best (n=4,η=1.4)	854	**0.3013**	0.3025	0.0006	0.636

Table 5.6: The STD performance on OOV terms with n-best predictions. The third column reports the number of predicted pronunciations. 'Pron.' denotes 'pronunciation'.

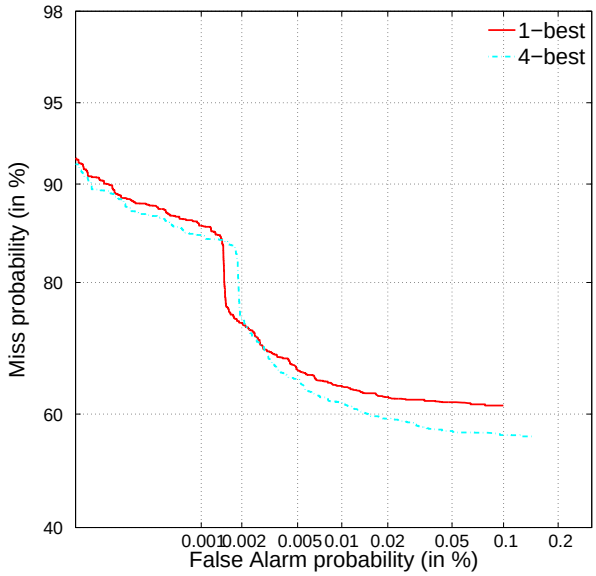

Figure 5.6: The DET curves of STD systems with the 1-best and 4-best pronunciation prediction. The confidence threshold η was set to 1.4.

Table 5.5 shows that using more pronunciations tends to result in more hits but trigger more false alarms. If the benefit from the miss rate reduction surpasses the expense from the false alarm rate increase, the overall performance is improved, otherwise the performance is decreased. This is why the performance improvement with the first few alternative pronunciations is so remarkable (e.g., in the case of 2-best prediction), but becomes marginal when more pronunciations are considered. With the 6-best prediction, those low-confidence pronunciations drove the performance down.

5.2.2 N-best prediction for INV terms

The n-best prediction can be applied to INV terms as well. The rationality is that additional pronunciations obtained from the n-best prediction can help address the pronunciation variation that commonly exists in spontaneous speech but is not specified in the dictionary.

Note that the AMI dictionary we used for experiments was carefully designed and contains alternative pronunciations, i.e., it is a multiple pronunciation dictionary. To test the contribution of the n-best prediction, we started from two *dictionary-based systems*: the first one used the original multiple pronunciation dictionary and the second one used a single pronunciation dictionary which was derived from the multiple pronunciation dictionary by removing the alternative pronunciations. The performance of these two systems is shown in Table 5.7, which shows that the system using the multiple pronunciation dictionary indeed outperformed the system using the single pronunciation system, confirming that considering alternative pronunciations does improve STD performance.

System name	Dictionary	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.singledct	single pronunciation	0.4612	0.4915	0.00005	0.485
phn.lt.multipledct	multiple pronunciation	0.4743	0.5058	0.00006	0.470

Table 5.7: The STD performance on INV terms using the single pronunciation dictionary and the multiple pronunciation dictionary.

Then we tested the *prediction-based systems*. We first generated n-best pronunciations for all INV terms using the joint-multigram model, and then applied these predicted pronunciations to perform STD. Experimental results are shown in Figure 5.7. For comparison, performance of the two dictionary-based systems are presented as well. We see that the n-best prediction-based systems outperformed the single pronunciation dictionary-based system, but was worse than the multiple pronunciation dictionary-based system.

Finally we used the n-best predicted pronunciations to augment the single pronunciation dictionary and the multiple pronunciation dictionary. Figure 5.8 reports the experimental results using the augmented single pronunciation dictionary, and Figure 5.9 reports the results using the augmented multiple pronunciation dictionary. It is interesting to see that adding the predicted pronunciations to the single pronunciation dictionary gave better STD performance, whereas the augmentation did not improve

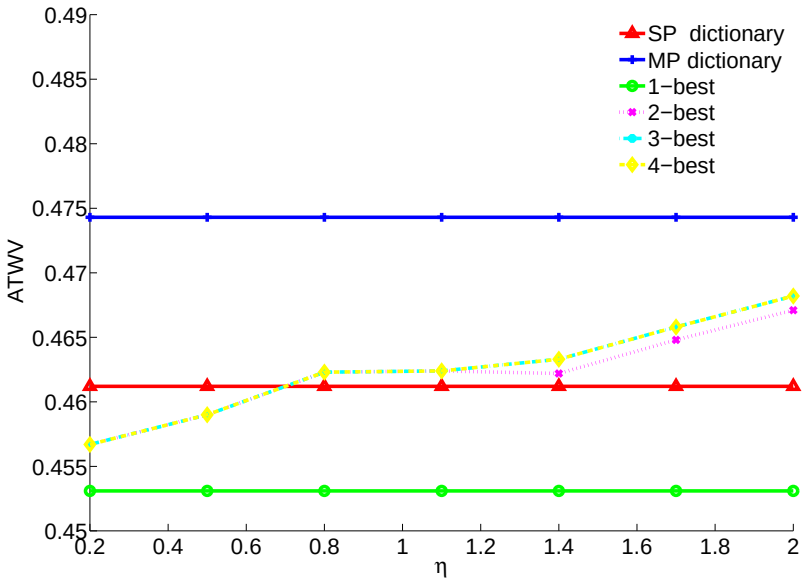

Figure 5.7: The STD performance on INV terms with n-best pronunciations predicted by the joint-multigram model. The confidence threshold η in the n-best prediction was set to 1.4. 'SP' denotes 'single pronunciation', and 'MP' denotes 'multiple pronunciation'.

the multiple pronunciation dictionary-based system, indicating that all informative pronunciations are already contained in the dictionary, and the predicted pronunciations did not provide extra information.

Summarising the results on INV and OOV terms, we conclude that the n-best prediction improves STD performance when the pronunciation of a term is unknown (in the case of OOV terms), or known a little (in the case of INV terms with a single pronunciation dictionary); if all the pronunciations are known (in the case of INV terms with a multiple pronunciation dictionary), the n-best prediction does not help.

5.3 Stochastic pronunciation model (SPM) for STD

An obvious shortcoming of the n-best prediction approach is that the maximum prediction number n and the confidence threshold η need to be determined empirically; moreover, the confidence scores of the predicted pronunciations have not yet been fully utilised.

To solve these problems, we merge the confidence of the predicted pronunciation and the confidence of the term detection into a *compound confidence*, based on which a *postponed decision* is made. With this approach, the parameter n and η are not

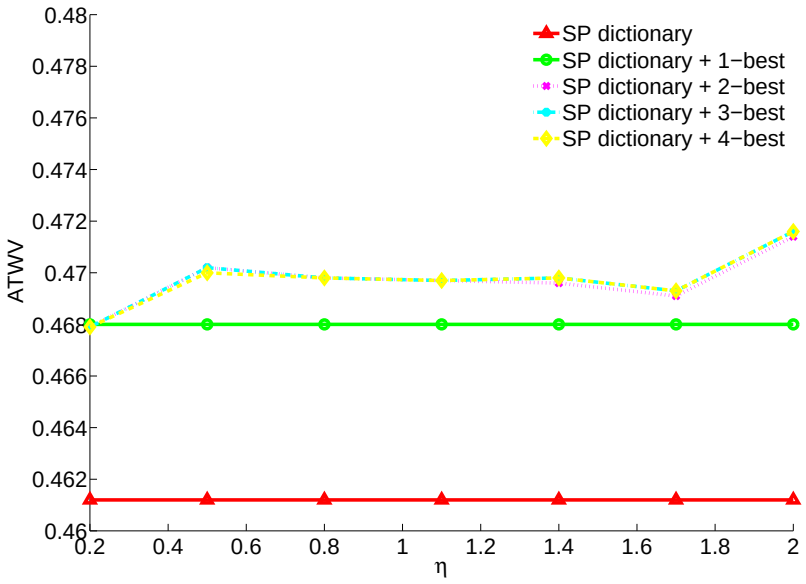

Figure 5.8: The STD performance on INV terms with the single pronunciation dictionary augmented by n-best pronunciations predicted by the joint-multigram model. 'SP' denotes 'single pronunciation'. The confidence threshold η in the n-best prediction was set to 1.4.

specified before hand; instead, all pronunciations are used for term search, and the hit/FA decision, based on the compound confidence, not only considers the reliability of the detection, but also the reliability of the pronunciation that the detection is based on.

We start from extending the definition of a detection d to become

$$d = (K, Q, s, c_{lattice}, c_{pron}, v_a, v_l, ...) \tag{5.15}$$

where the pronunciation Q and its confidence c_{pron} are introduced, and $c_{lattice}$ is the lattice-based confidence defined by Equation 3.19 that is reproduced here for convenience.

$$\begin{aligned} c_{lattice}(d) &= P(K_{t_1}^{t_2}, Q(d)|O_1^T) && (5.16) \\ &= \sum_{C_K} \frac{p(O_1^T|C_K, K_{t_1}^{t_2}, Q(d))P(C_K, K_{t_1}^{t_2}, Q(d))}{p(O_1^T)} && (5.17) \end{aligned}$$

The newly introduced attribute c_{pron} of d represents the confidence of the pronunciation Q on which the detection d has been found, and is defined as the posterior

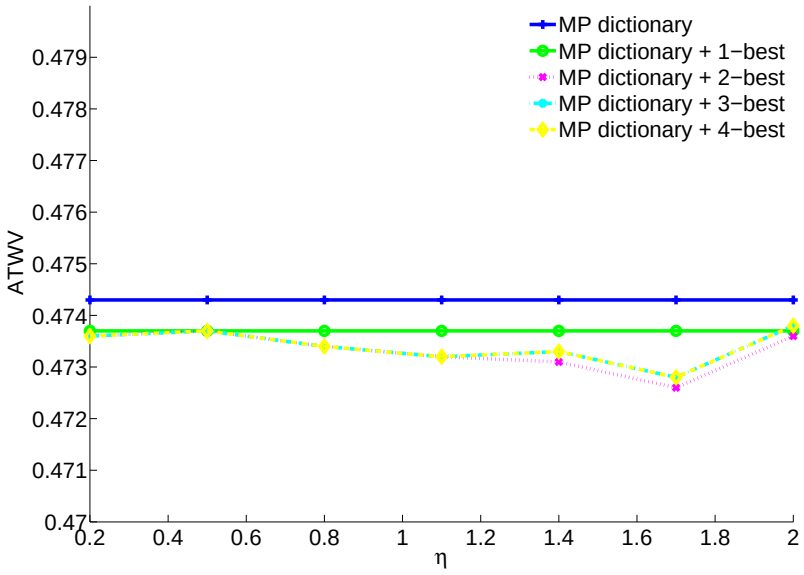

Figure 5.9: The STD performance on INV terms with the multiple pronunciation dictionary augmented by n-best pronunciations predicted by the joint-multigram model. 'MP' denotes 'multiple pronunciation'. The confidence threshold η in the n-best prediction was set to 1.4.

probability of Q given the term K:

$$c_{pron}(d) = P(Q|K). \tag{5.18}$$

We denote a model describing $P(Q|K)$ a *stochastic pronunciation model (SPM)* for STD. According to Equation 4.26, the joint-multigram model can readily generate $P(Q|K)$, so it is an ideal SPM.

With the pronunciation confidence c_{pron} from the SPM and the detection confidence $c_{lattice}$ from the lattice, we can compute the compound confidence $c(d)$ so that the postponed decision can be made. The basic idea is to estimate the joint probability of the term and the pronunciation detected in the lattice, i.e.,$P(Q,K|O)$. Assuming unified prior probabilities $P(Q)$ and $P(K)$, we have

$$
\begin{aligned}
P(Q,K|O) &\propto \frac{P(Q)}{P(K)}P(Q,K|O) && (5.19)\\
&= P(K|Q,O)P(Q|O)\frac{P(Q)}{P(K)} && (5.20)\\
&= P(K|Q)P(Q|O)\frac{P(Q)}{P(K)} && (5.21)\\
&= P(Q|K)P(Q|O) && (5.22)
\end{aligned}
$$

where we have assumed that the term K is independent of the speech O conditioned on the pronunciation Q when deriving Equation 5.20 to Equation 5.21.

In practice, a scale factor γ can be introduced to manage the contribution of the elementary probabilities $P(Q|K)$ and $P(Q|O)$ in Equation 5.22; further more, we find exponential escalation on the elementary probabilities would improve the performance; finally, the resulted score is mapped by logarithm. This gives rise to a compound confidence $c(d)$ as follows,

$$c(d) = log\{(e^{P(Q|O)})^{1-\gamma}(e^{P(Q|K)})^{\gamma}\}. \quad (5.23)$$

By simple arrangement, this leads to a linear interpolation of the detection confidence $c_{lattice}$ and the pronunciation confidence c_{pron} shown as follows,

$$c(d) = (1-\gamma)c_{lattice}(d) + \gamma c_{pron}(d) \quad (5.24)$$

where γ is an interpolation factor.

Theoretically, the term detector can take all the pronunciations from the SPM for search. In practice, however, the number of pronunciations that can be processed is limited by the computing resource. In our experiments, 50-best pronunciations were generated for each term and searched for within the lattices. The experimental results are shown in Figure 5.10, where the ATWV is plotted against the interpolation factor γ.

In practice, γ was chosen to maximise the STD performance on the development set, which gave γ=0.7 in our experiments. Applying this optimal value to conduct the evaluation, we obtained the results shown in Table 5.8. For comparison, results of the systems using 1-best prediction and n-best prediction are listed as well. Figure 5.11 shows the DET curves of the STD systems with 1-best prediction, n-best prediction and the SPM.

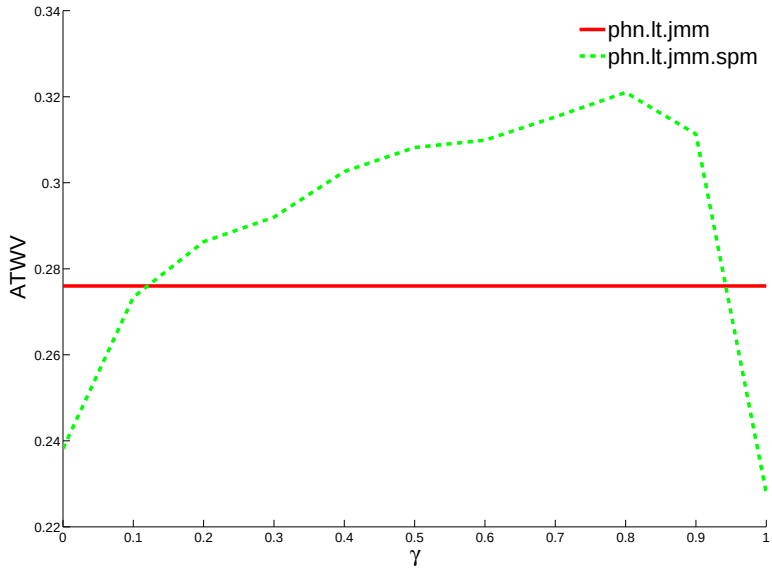

Figure 5.10: The STD performance on OOV terms with the joint-multigram model-based SPM.

System name	Prediction	#Pron.	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.jmm	1-best	484	0.2761	0.2770	0.00006	0.667
phn.lt.jmm.nbest	n-best (n=4,θ=1.4)	854	0.3013	0.3025	0.00006	0.636
phn.lt.jmm.spm	50-best (γ=0.7)	20877	**0.3153**	0.3303	0.00008	0.604

Table 5.8: The STD performance on OOV terms with the joint-multigram model-based SPM. The third column reports the number of predicted pronunciations.

The results shown above confirm that the SPM-based approach does substantially improve performance of STD systems. A pairwise t-test shows that the SPM-based system outperformed the 1-best prediction-based system significantly ($p \approx 0.005$), though not significantly over the n-best prediction-based system ($p \approx 0.2$).

An interesting observation is that the DET curve of the SPM-based system is lower than that of the 1-best prediction-based system, indicating that false alarms introduced by the multiple predicted pronunciations can be suppressed effectively by the SPM and the postponed decision.

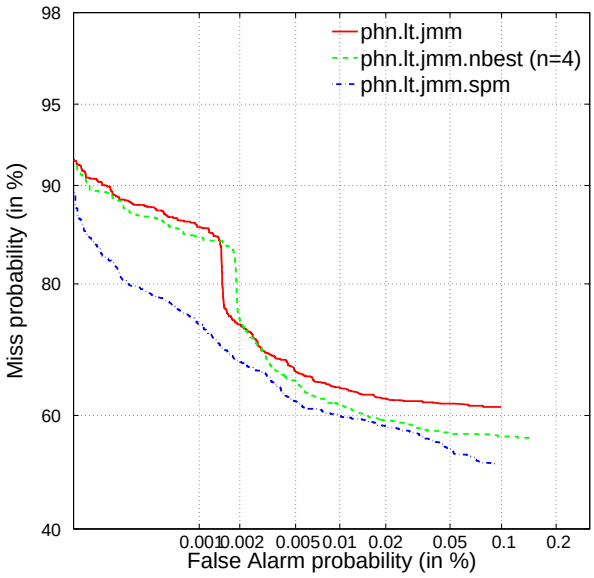

Figure 5.11: The DET curves of STD systems with 1-best prediction, n-best prediction and the joint-multigram model-based SPM.

5.4 Soft match

In general, the SPM-based approach belongs to a family of techniques used to deal with variation in pattern search, known as *query expansion*. The basic idea is that the search pattern can be expanded into a set of similar patterns so that variations on the search pattern can be found. Query expansion has been proposed for spoken document retrieval based on the edit distance [Wechsler and Schäuble, 1995] or acoustic confusion [Logan et al., 2005]. To the author's best knowledge, the joint-multigram model-based query expansion has not previously been reported, especially for STD tasks.

Another approach to compensate the pronunciation variation for STD is *soft match*. Different from query expansion which constructs similar query terms, soft match takes alternative pronunciations by allowing some mismatches between the canonical pronunciation and the real pronunciation of a detection. The mismatch cost can be calculated based on the edit distance [James and Young, 1994; Thambiratnam and Sridharan, 2005; Szöke et al., 2005a; Bosch et al., 2006; Miller et al., 2007] or acoustic confusion [Wechsler et al., 1998; Srinivasan and Petkovic, 2000; Audhkhasi and Verma, 2007].

Note that query expansion can be applied to either a word-based system [Logan and Thong, 2002] or a phoneme-based system [Logan et al., 2005], while soft match

can be applied to phoneme-based systems only.

5.4.1 Confusion matrix-based soft match

For an STD system based on phoneme lattices, pronunciation variation can be largely compensated for by alternative paths in the lattice. If the lattice is dense enough, canonical pronunciations are much likely to be involved. In that case, exact match is enough to catch the term occurrence [Szöke et al., 2005a]. In our experiments, however, the lattices were heavily pruned by a 6-gram phoneme LM, which means soft match could be desirable.

For convenience, we use the term *match confidence* to specify the impact of an inexact match. Since acoustic confusion provides more information than edit distance, we chose to derive the match confidence from a confusion matrix[1]. The confusion matrix describes the probability that a phoneme is recognised as another phoneme by an ASR system, and can be obtained by aligning the recognition result on the development set and the reference phoneme transcript. These probabilities are described by the following equations:

$$P_s(q'|q) = \frac{N_s(q',q)}{N(q)} \tag{5.25}$$

$$P_d(q) = \frac{N_d(q)}{N(q)} \tag{5.26}$$

$$P_i(q) = \frac{N_i(q)}{N(q)} \tag{5.27}$$

where $P_s(q'|q)$ is the probability that q is recognised as q', and $P_d(q)$ and $P_i(q)$ are the probabilities that q is deleted from and inserted into the canonical transcription respectively. In addition, $N(q)$ denotes the number of occurrences of q in the reference transcript, and $N_s(q',q)$, $N_d(q)$, $N_i(q)$ denote the number of aligned pairs of (q',q), the number of deletions of q and the number of insertions of q respectively.

The substitution/deletion/insertion probability is then used as the match confidence in term search, written as:

[1]The edit distance-based confidence can be regarded as a special case of the confusion matrix-based confidence where the non-zero elements of the confusion matrix are set to be a uniform value.

$$c_s(e) = P_s(e_i|e_j) \tag{5.28}$$
$$c_d(e) = P_d(e_i) \tag{5.29}$$
$$c_i(e) = P_i(e_j) \tag{5.30}$$
$$\tag{5.31}$$

where c_s, c_d, c_i are the match confidence for substitution, deletion and insertion respectively, and $e = (e_i, e_j) = (q', q)$ denotes a match of q to q'. Note that q and q' can be empty, in which case e denotes an insertion or a deletion. An interesting point is that even if q and q' are identical, i.e., e is an exact match, the match still possesses a non-trivial confidence, indicating some intrinsic uncertainty existing in exact match.

Now we define the confidence of a match of two pronunciations Q_d and Q_l as the accumulation of the confidence of all the phoneme matches in the alignment of Q_d and Q_l, denoted as $c_{match}(Q_d, Q_l)$. Thus we have

$$c_{match}(Q_d, Q_l) = \prod_i c(e(i)) \quad c \in \{c_s, c_d, c_i\} \tag{5.32}$$

where $e(i)$ is the match of the i-th phoneme pair in the alignment of Q_d and Q_l.

The soft match-based lattice search can be implemented by a small extension to the searching algorithm for the exact match. The only change is to allow unmatched paths in the lattice to be extended, subject to some constraints. In our implementation, the only constraint is the maximum number of unmatched phonemes between the canonical pronunciation and the detected pronunciation, denoted as n_m.

To derive the decision strategy, we first extend the definition of detection d by adding the canonical and detected pronunciations, giving

$$d = (K, Q_d, Q_l, s, c_{lattice}, v_a, v_l, c_{match}, ...) \tag{5.33}$$

where Q_d is the canonical pronunciation of the term and Q_l is the real pronunciation of the detection, and c_{match} represents the match confidence we defined in Equation 5.32.

As in the case of SPM, we compute the compound confidence $c(d)$ as the interpolation of the lattice-based confidence $c_{lattice}$ and the match confidence c_{match}, that is

$$c(d) = (1-\nu)c_{lattice}(d) + \nu c_{match}(d) \tag{5.34}$$

where ν is the interpolation factor.

5.4.2 Experimental results

To perform soft match-based STD, we first of all need to generate the confusion matrix. To do that, we conducted phoneme recognition on the development set using the same decoder that was used for lattice generation, and then aligned the decoding result with the reference transcript. Finally the confusion matrix was computed from the alignment according to Equation 5.25-5.27.

The STD experiments on OOV terms were conducted with the joint-multigram model-based 1-best pronunciation prediction. To prevent a flood of putative detections, the maximum number of mismatches n_m was confined to 0, 1 or 2. Figure 5.12 shows the results with soft match when the interpolation factor ν changing from 0 to 1.

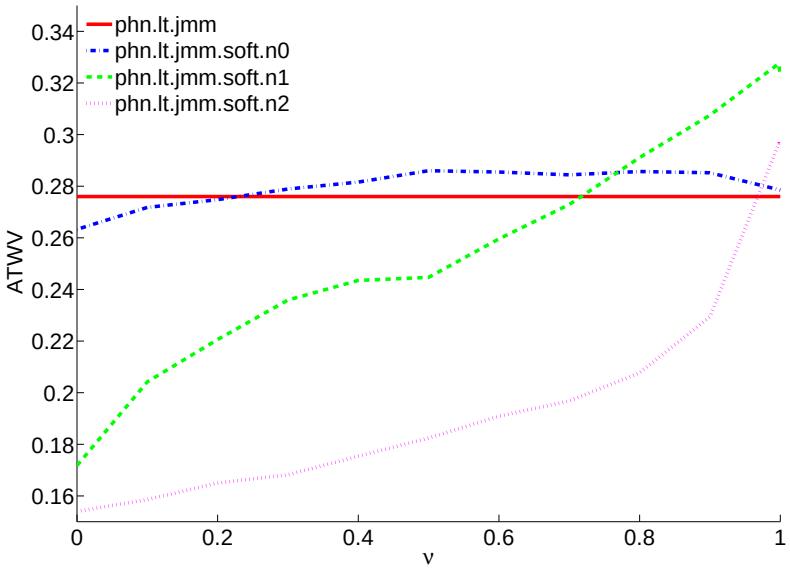

Figure 5.12: The performance of the STD system on OOV terms with soft match, when the interpolation factor ν changing from 0 to 1. The maximum number of mismatches n_m was set from 0 to 2, corresponding to the three systems $phn.lt.jmm.soft.n0$, $phn.lt.jmm.soft.n1$ and $phn.lt.jmm.soft.n2$.

Table 5.9 reports the experimental results of the soft match-based system, with optimal values of ν that were obtained by optimising system performance on the development set. We can see that the soft match-based system allowing at most 1 mismatch gave the best result. A pairwise t-test shows that this result is significantly better than the result based on exact match ($p < 0.01$).

An interesting observation is that the performance of the exact match-based system

was improved by using the match confidence, suggesting that some prior knowledge of the reliability of a match was introduced. Another observation is that allowing some mismatches (e.g., n_m=2) might increase the max-ATWV, but did not necessarily increase the ATWV, which indicates that the tuned parameters on the development set did not apply to the evaluation well. This might be because the large number of false alarms introduced by setting n_m to a large value makes it hard to choose a reliable confidence threshold. Finally, we find that the optimal values of ν are very close to 1 in the cases of $n_m > 0$, which indicates that the optimal decision had assigned a high priority to the matching confidence, and the optimal ν tuned on the development set might be suboptimal for evaluation.

System name	n_m/ν	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.jmm	-	0.2761	0.2770	0.00006	0.667
phn.lt.jmm.soft.n0	0/0.700	0.2844	0.2862	0.00003	0.682
phn.lt.jmm.soft.n1	1/0.998	**0.3275**	0.3300	0.00004	0.637
phn.lt.jmm.soft.n2	2/0.998	0.2965	0.3425	0.00003	0.676

Table 5.9: The performance of STD systems using soft match, with the interpolation factor ν setting to the optimal values tuned on the development set. The best result is given in bold face.

More properties of the soft match-based detection can be observed from the DET curves shown in Figure 5.13. The first observation is that the soft match-based system performed better than the exact match-based system when the false alarm rate is high, but worse when the false alarm rate is low. The second observation is that the SPM-based system performed the best in the whole operating region except where the false alarm rate is relatively high. The different behaviour of the SPM and soft match -based approaches will be discussed in the following section.

5.5 SPM and soft match

In this section, we compare the SPM and soft match as different means of uncertainty treatment for STD, and combine these two approaches for further performance improvement.

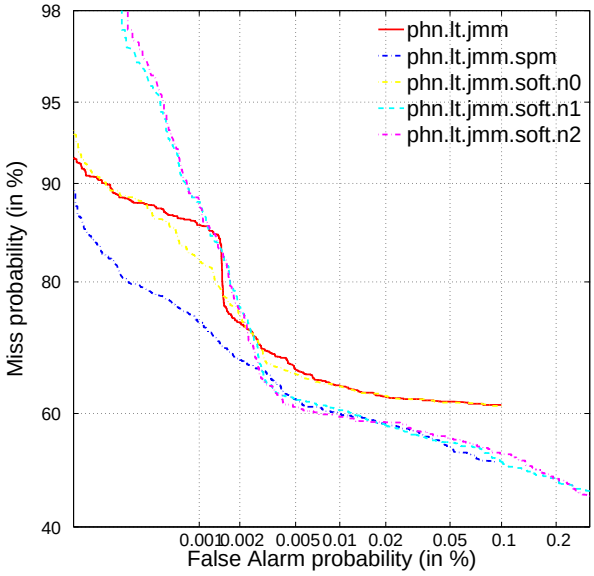

Figure 5.13: The DET curves of the soft match-based systems with the maximum number of mismatches n_m set from 0 to 2. The interpolation factor ν was set to the optimal value obtained by tuning on the development set.

5.5.1 Dealing with pronunciation uncertainty

We have utilised three approaches to deal with the pronunciation uncertainty in STD: lattice-based representation, SPM and soft match; all these techniques attempt to match the search terms against the ASR results that otherwise do not match.

At the first sight, these approaches differ only in the objects they work with: the lattice-based approach works on recognition results, and the SPM works on search terms, while the soft match works on the searching process. However, we will see that the SPM is fundamentally different from the other two approaches with respect to the idea of uncertainty treatment.

We know that the lattice representation aims to compensate recognition errors by alternative hypotheses, while the soft match tries to recover the pronunciations that are pruned away in decoding. Although somewhat different, these two approaches hold the same *canonical pronunciation assumption*, i.e., the pronunciation of the search term is correct, therefore we just need to manipulate the recognition results or the searching process to match the canonical pronunciation. The SPM, on the contrary, holds a *stochastic pronunciation assumption*, i.e., the pronunciation of a term is undetermined, thus any of the possible pronunciations should be considered when searching a term.

The canonical pronunciation assumption is rational when the dictionary has been thoroughly designed and contains all possible pronunciations. For OOV terms, however, the stochastic pronunciation assumption is more reasonable, since the correct pronunciations are actually unknown, and the LTS model used to predict these pronunciations is itself stochastic.

In addition, these models reflect distinct properties in representing alternative pronunciations. The lattice representation applies both acoustic and LM constraints and thus is the most restricted, which usually leads to reasonable alternative pronunciations, but tends to prune away some occurrences of OOV terms. The soft match uses a phoneme confusion matrix, thus is purely acoustically driven and rather flexible. The SPM applies phonological constrains represented by the joint-multigram model, so is purely phonologically driven and more constrained. This analysis explains why the SPM-based approach achieved better results than the soft match-based approach: the SPM makes use of the phonological constraints so the generated pronunciations tend to follow phonological rules, while the soft match allows all possible phoneme substitutions, thus may predict some pronunciations that do not exist in reality.

In our experiments, the lattice-based approach is indispensable due to its ability to suppress false alarms; the SPM is important as well since we need deal with OOV terms which have no canonical pronunciations; the soft match is also desirable since it recovers some occurrences that are pruned away in decoding thus missed in the lattice. It would be ideal if we can find a way to integrate all of these approaches in a single framework.

5.5.2 Computation workload

In Section 3.1.3, we have mentioned that the computation complexity of the dictionary tree-based exact match is $O(N \times (\beta \times B \times D)^L)$, in which N is the number of nodes of the lattice, B is the average number of fan-out arcs of the nodes, D and L are the average number of fan-out arcs and the average depth of the dictionary tree respectively, and β is the probability of successful match in lattice search. Following this representation, the additional computation with SPM can be thought to be caused by the increased fan-out D and depth L of the dictionary tree, while the workload introduced by soft match can be thought to be caused by the increased match probability β.

In order to examine the computation workload in practice, we chose 500 utterances randomly from the evaluation set, and performed the OOV STD with 1-best

exact match, SPM (50-best, exact match) and soft match (maximum one substitution) respectively. The experiments with the three methods were conducted in serial within the same computational environment (Dual Intel Xeon E5320 , 12GB memory), without any other resource-intensive jobs existing.

Algorithm	N	# pron.	B	L	β	Exec. time [s]
1-best exact match	1.75	484	0.999492	6.017660	0.04	287
SPM	1.75	20877	0.999981	6.321731	0.03	331
soft match	1.75	484	0.999492	6.017660	0.13	1616

Table 5.10: The computation workload when performing OOV STD on 500 utterances, with 1-best exact match, SPM and soft match. '# pron.' denotes the number of pronunciations that are searched for. 'Exec. time' is the time cost in seconds when conducting the lattice search.

The experimental results are shown in Table 5.10. N was obtained by examining the lattices of the 500 utterances; B and L were obtained by examining the dictionary tress of the 1-best and 50-best pronunciations, for 1-best match and SPM respectively. To obtain β, we recorded the phoneme comparison operations in lattice search, and estimated β as the proportion of successful matches in all comparisons.

From the results in Table 5.10, we first see that both SPM and soft match introduced additional workload and slowed down the lattice search. This is necessary expense for uncertainty treatment though. Secondly, we observe that with SPM, B and L were increased; with soft match, β was increased. This is consistent with our analysis. Finally, we find the workload caused by SPM is rather insignificant ($\approx 15.5\%$), which should be attributed to the efficient data sharing with dictionary trees; by contrast, the workload caused by soft match was quite significant (5 times), probably due to the fact that we allowed substitutions between any two phonemes. This suggests that SPM tends to be more efficient than soft match when dealing with pronunciation uncertainty.

5.5.3 SPM-based soft match with linear interpolation

As we have mentioned, SPM and soft match deal with pronunciation uncertainty from different aspects, and therefore can be combined to give a more comprehensive uncertainty treatment.

The first approach we tried to integrate the SPM and soft match is an SPM-based

soft match approach. In this approach, we employ the SPM to predict multiple pronunciations, and then apply soft match to search for the enquiry terms, assuming that the predicted pronunciations are correct.

We follow the same idea of compound confidence and postponed decision. For that, we first extend the definition of detection d to include extra information from the SPM and soft match, giving

$$d = (K, Q_d, Q_l, s, c_{lattice}, c_{pron}, c_{match}, v_a, v_l, ...) \tag{5.35}$$

where Q_d and Q_l are the canonical and real pronunciation, and c_{match} and c_{pron} are the match confidence and pronunciation confidence respectively.

The compound confidence is calculated as an interpolation of the lattice-based confidence $c_{lattice}$, the match confidence c_{match} and the pronunciation confidence c_{pron}, giving

$$c(d) = (1-\gamma-\nu)c_{lattice}(d) + \gamma c_{pron}(d) + \nu c_{match}(d) \tag{5.36}$$

where γ and ν are two interpolation factors.

We conducted the experiments with γ and ν optimised on the development set. The maximum number of mismatches n_m was set equal or less than 1 as $n_m > 1$ led to too many putative detections but no performance improvement on the development set. The results are reported in Table 5.11.

System name	n_m/γ/ν	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.jmm.spm	-/0.700/-	0.3153	0.3303	0.00008	0.604
phn.lt.jmm.spm.soft.n0	0/0.800/0.01000	**0.3221**	0.3358	0.00008	0.594
phn.lt.jmm.spm.soft.n1	1/0.999/0.00099	0.1790	0.2440	0.00012	0.700

Table 5.11: The STD performance with SPM-based soft match. The best result is shown in bold face.

From the above results, we first notice that, if no mismatch was allowed ($n_m = 0$), taking account of the match confidence provided a small performance improvement for the SPM-based system; however, when mismatches were allowed ($n_m > 0$), the performance of the SPM-based system was substantially reduced. In other words, the SPM-based soft match did not work well.

Examining the results in Table 5.11, we found that the optimal values of the interpolation factors are rather odd in the SPM-based systems applying soft match. It seems

that the optimal system was trying to make a step-wise decision: it first looked at the match confidence to choose detections that were exactly matched, and then looked at the pronunciation confidence to choose detections that were found according to the most probable pronunciations. In this case, the lattice-based confidence was ignored to a large extent. This behaviour is somewhat understandable if we notice that the decision maker has to reduce flood of false alarms introduced by the two variation compensation approaches. We hypothesise that the linear classifier represented by Equation 5.36 is unable to deal with the decision problem based on multiple confidence. This motivated us to look for non-linear decision approaches as discussed in the next chapter.

5.5.4 Detection combination for SPM and soft match

The second approach we tried to integrate the SPM and soft match is the detection combination that was discussed in Section 5.1.4. With this approach, the SPM-based and soft match-based systems perform the term detection separately, and then the detections from both systems are merged into the final results. We combined the best SPM-based system *phn.lt.jmm.spm.soft.n0* and the best soft match-based system *phn.lt.jmm.soft.n1*, getting the results shown in Table 5.12. The DET curve of the combined system is shown in Figure 5.14.

System name	ATWV	max-ATWV	P(FA)	P(Miss)
phn.lt.jmm.spm.soft.n0	0.3221	0.3358	0.00008	0.594
phn.lt.jmm.soft.n1	0.3275	0.3300	0.00004	0.637
phn.lt.jmm.spm.soft.n0+phn.lt.jmm.soft.n1	**0.3427**	0.3534	0.00004	0.613

Table 5.12: The STD results on OOV terms by merging detections from the SPM-based system and the soft match-based system. The best result arises from the combined system *phn.lt.jmm.spm.soft.n0+phn.lt.jmm.soft.n1*, as presented in bold face.

The results shown above confirm that the detection combination indeed improved the STD performance, although the improvement over the soft match-based system is not significant ($p \approx 0.2$). From the DET curves, we find that the combined system outperformed each individual system if the false alarm rate is high. Due to the low accuracy of the soft match-based system, the combined system performed worse than the SPM-based system if the false alarm rate is low.

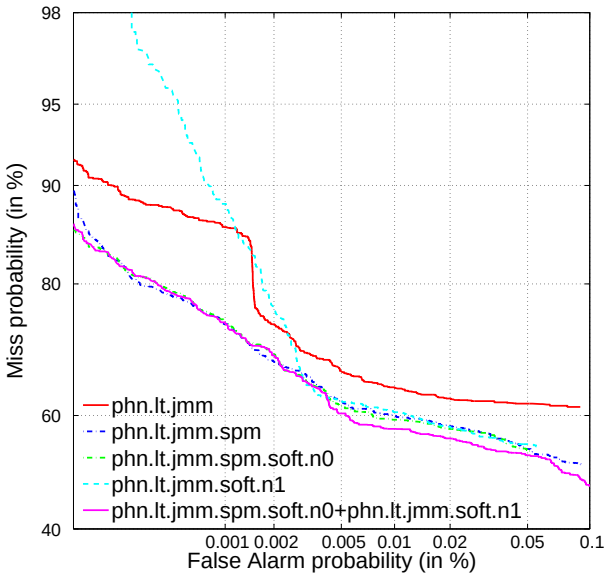

Figure 5.14: The DET curves of the STD systems based on SPM, soft match and their detection combination. The combined system is named as *phn.lt.jmm.spm.soft.n0+phn.lt.jmm.soft.n1*.

5.6 Summary

In this chapter we first applied the joint-multigram model to predict pronunciations for OOV terms, and then extended the 1-best prediction to an n-best prediction. Based on the n-best prediction, a stochastic pronunciation model (SPM) was introduced, and was compared with the soft match-based approach. We conclude that the joint-multigram model gave significantly better performance for STD on OOV terms than the CART, and the SPM further improved the performance considerably. The detection combination is a feasible approach to combine the SPM and soft match, and gave better results than systems based on individual techniques.

A critical problem we face with the SPM and soft match is that the linear interpolation we used to integrate various confidence measures ($c_{lattice}$, c_{pron}, c_{match}) might be suboptimal. These confidence measures are derived from different parts of the detection system and reflect different properties of the putative detections, which makes their distributions highly diverse in ranges and forms. This diversity suggests that the optimal function for multiple confidence integration might be rather complex. Moreover, the confidence measures we applied so far are not discriminative in the sense of hit/FA classification, which leads to suboptimal decisions. In the next chapter, we ap-

ply various discriminative models to normalise these confidence measures to posterior probabilities, based on which a discriminative decision is made.

Chapter 6

Discriminative decision making

Within an STD system, the decision maker plays an important role in ascertaining reliable detections according to some decision strategies. The simplest decision strategy, as we applied to the baseline system, is term-independent, which is not suitable for OOV term detection because of the high property diversity among OOV terms. An ideal decision strategy for OOV term detection should take account of term-dependent factors so that the OOV diversity could be compensated for. Second, as we have seen in the previous chapter, linear interpolation is not suitable to integrate diverse confidence measures when applying SPM or soft match to treat OOV uncertainty. We need a general framework to combine various confidence measures and integrate term-dependent factors to improve quality of the decision making.

In this chapter, we propose a discriminative decision strategy which integrates various decision factors into a classification posterior probability with some discriminative models. With this approach, multiple confidence measures and various term-dependent factors are integrated to make a term-dependent discriminative decision that leads to minimum decision cost.

6.1 Discriminative decision making

6.1.1 Discriminative confidence and decision

Let us re-examine the decision strategy formulated by Equation 3.24. For convenience, it is reproduced as follows:

$$assert(d) = \left\{ \begin{array}{ll} 1 & if\ c(d) \geq \theta \\ 0 & if\ c(d) < \theta \end{array} \right\} \tag{6.1}$$

where θ is a pre-defined threshold, and $c(d)$ is the confidence of a detection d defined as follows,

$$d = (K, Q_d, Q_l, s, v_a, v_l, c_{lattice}, c_{pron}, c_{match}, ...) \tag{6.2}$$

in which K is the detected term, Q_d and Q_l are the dictionary pronunciation and detected pronunciation respectively, s denotes the speech segment, and v_a, v_l, $c_{lattice}$, c_{pron}, c_{match} are the acoustic likelihood, language model score, lattice-based confidence, pronunciation confidence and match confidence, respectively. Note that applying the confidence normalisation does not change the general form of the decision strategy given above, if we treat $c(d)$ as a normalised confidence.

The question is: Is the decision strategy of Equation 6.1 optimal in terms of decision cost? To get the answer, we cast the problem of hit/FA decision to a problem of binary classification, for which the goal is to assign a detection d into either one of two classes: hits C_{hit} and false alarms C_{FA}. According to decision theory, the optimal classification strategy in terms of classification cost is to assign d to C_{hit} if and only if

$$P(C_{hit}|d) \geq \beta P(C_{FA}|d) \tag{6.3}$$

where β is a scale factor to weight hits and false alarms. Considering

$$P(C_{hit}|d) + P(C_{FA}|d) = 1 \tag{6.4}$$

we get the optimal classification strategy:

$$classification(d) = \left\{ \begin{array}{ll} C_{hit} & if\ P(C_{hit}|d) \geq \dfrac{\beta}{1+\beta} \\ C_{FA} & if\ P(C_{hit}|d) < \dfrac{\beta}{1+\beta} \end{array} \right\} \tag{6.5}$$

Looking back at the decision strategy of Equation 6.1, we see that the decision is optimal if and only if the confidence $c(d)$ holds a close relationship with the classification posterior probability $P(C_{hit}|d)$, given by

$$c(d) \geq \theta \Longleftrightarrow P(C_{hit}|d) \geq \frac{\beta}{1+\beta}. \tag{6.6}$$

We call a confidence measure holding this relationship a *discriminative confidence*, and a decision based on a discriminative confidence a *discriminative decision*. A typical discriminative confidence is the classification posterior probability, formally represented by c_{disc} and given by

$$c_{disc}(d) = P(C_{hit}|d). \tag{6.7}$$

6.1.2 Review of lattice-based confidence

Unfortunately, the lattice-based confidence $c_{lattice}$ we utilised so far is not discriminative. Recall that $c_{lattice}$ represents the detection posterior probability $P(K_{t_1}^{t_2}|O)$ which does not hold a close relationship with $P(C_{hit}|d)$. Figure 6.1 shows an example where the threshold on $P(C_{hit}|d)$ that leads to an optimal classification does not correspond to a threshold on $c_{lattice}$. This indicates that the lattice-based confidence is not discriminative.

6.1.3 Confidence normalisation and discriminative decision

Confidence normalisation and discriminative decision making are two approaches to improve the decision quality of an STD system by increasing the discriminative power of the confidence measure; however they arise from different ideas: confidence normalisation originates from maximising the evaluation metric ATWV, while discriminative decision making arises from minimising the decision cost. Therefore, they are probably complementary and can be applied together.

Moreover, examining the normalisation formulas in Equation 5.6 and Equation 5.8, we find that the only valid choice of the confidence $c(d)$ for these formulas is the classification posterior probability $P(C_{hit}|d)$. This indicates that the discriminative confidence and confidence normalisation are intrinsically consistent.

This analysis suggests that these two techniques can be combined, which leads to a term-dependent normalised discriminative confidence. The underlying idea is to apply

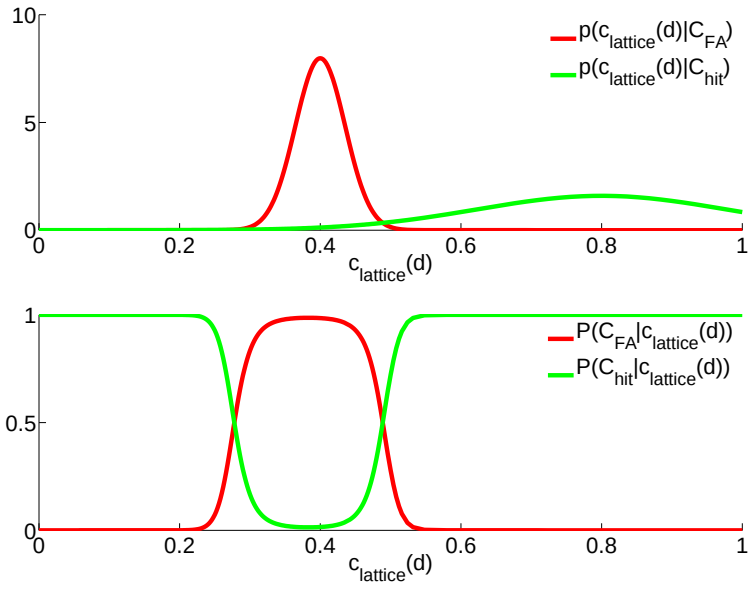

Figure 6.1: An example that the lattice-based confidence is not closely related to the classification posterior probability. In the top diagram, the probability distribution over the lattice-based confidence $c_{lattice}$ is plotted for hits (C_{hit}) and false alarms (C_{FA}), and in the bottom diagram, the corresponding classification posterior probability distribution is plotted.

the term-dependent discrimination to minimise the decision cost and apply the term-dependent confidence normalisation to optimise the STD performance with respect to ATWV.

6.2 Mapping to discriminative confidence

6.2.1 Short mapping and long mapping

We can derive the discriminative confidence c_{disc} by either converting the lattice-based confidence with a short mapping g,

$$g : c_{lattice}(d) \longrightarrow c_{disc}(d) \tag{6.8}$$

or considering more decision factors with a long mapping f:

$$f : (c_{lattice}(d), c_0, c_1, ...) \longrightarrow c_{disc}(d) \tag{6.9}$$

where c_0, c_1,... denote decision factors such as the acoustic likelihood, LM score, match confidence or pronunciation confidence. Compared to a short mapping, a long mapping may discover some potential dependence between the discriminative confidence and various *raw* informative factors.

6.2.2 Model-based discriminative confidence estimation

It is difficult to devise a formula to represent the mapping from decision factors to the discriminative confidence, as we usually do not know which factors are informative, and how to choose a function to combine these factors. Therefore, we apply a model-based approach, which represents the discriminative mapping by a probabilistic model that is learnt from a set of training exemplars by some machine learning techniques. Although it is often demanding in computation, the model-based approach is easy to design and implement. Moreover, if the training data are plentiful, the probabilistic model approaches the real mapping function.

We tested two discriminative models: a multiple layer perceptron (MLP) [Mathan and Miclet, 1991] and a support vector machine (SVM) [Zhang and Rudnicky, 2001]. Both of them estimate the classification posterior probability $P(C_{hit}|d)$ with unlimited accuracy given unlimited training data.

6.2.3 Model-based discriminative confidence for OOV terms

The effect of the model-based discriminative confidence is rather complex when it is applied to OOV terms. On one hand, OOV terms tend to be weakly modelled by the acoustic and language models, which causes the model-derived lattice-based confidence to be less discriminative. The discriminative approach is able to improve the discriminative power of confidence measures and hence is highly desirable for OOV terms. On the other hand, learning discriminative models for OOV terms tends to suffer from data sparsity, giving unreliable models and unreliable confidence scores.

In summary, a discriminative confidence has a two-fold effect for OOV term detection: it provides more discriminative power but tends to be unreliable. We need to examine which effect is dominant in practice.

6.3 Discriminative decision based on short mappings

A short mapping converts the lattice-based confidence to the discriminative confidence without considering any other decision factors, as formulated by Equation 6.8. Basically, this mapping represents a (possibly non-linear) scaling function that normalises the lattice-based confidence according to some statistical knowledge learnt in model training. In this section, we describe the implementation and experimental results of the discriminative decision based on short mappings.

6.3.1 Training data

To train a discriminative model, we started by collecting positive and negative training exemplars. Specifically, we collected the putative detections that were generated by performing STD on the development set, and then labelled each detection as a hit or a false alarm. Afterwards, the hits were used as positive exemplars and the false alarms were used as negative exemplars to train the model. To obtain more training exemplars, we collected exemplars from both the INV terms and OOV terms for training, and exemplars from the OOV terms for cross validation.

6.3.2 Discriminative model training

We chose a MLP and a SVM as the discriminative model. Both of them output classification posterior probabilities and are widely used for classification. To train the MLP, we used a tool developed by ourselves; to train the SVM, we used a public tool called LIBSVM [Chang and Lin, 2001] from the National Taiwan University. The details of the training process for each model are presented as follows.

Multiple layer perceptron (MLP)

A standard 3-layer MLP was trained with the training exemplars. The structure is comprised of an input layer and a hidden layer with a sigmoid activation, and an output layer with a soft-max activation, as shown in Figure 6.2.

We adopted the standard error back-propagation (BP) algorithm to train the model, as described by Bishop [1995]. The optimal number of hidden units were chosen to minimise the classification errors on the validation set, which is 30 is our experiments. We denote the system based on the MLP as *phn.ld.mlp* for INV terms and *phn.ld.mlp.jmm* for OOV terms.

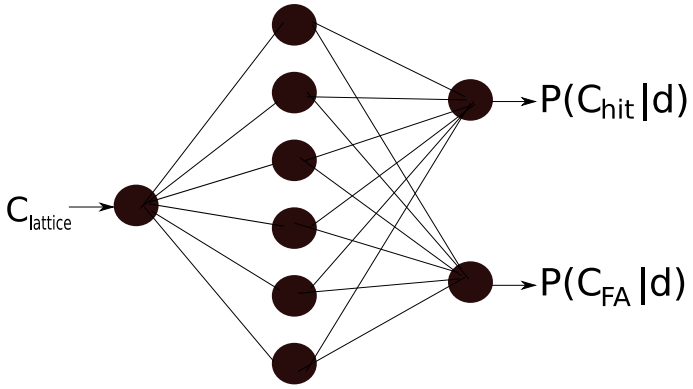

Figure 6.2: The MLP structure that represents the short mapping that converts the lattice-based confidence to the discriminative confidence.

Support vector machine (SVM)

The SVM was trained from the same training data with a toolkit called LIBSVM from the National Taiwan University [Chang and Lin, 2001]. The radial basis function was selected as the kernel, and parameters, including the error penalty C for classification and the radius scale γ for the kernal, were optimised to maximise the classification performance on the validation set, giving $C = 32$ and $\gamma = 0.5$ in our experiments. We denote the system based on the SVM as *phn.ld.svm* for INV terms and *phn.ld.svm.jmm* for OOV terms,

6.3.3 Handling data imbalance

A critical problem in model training is that there were more negative exemplars (false alarms) than positive exemplars (hits) in the training set, which made the trained model biased towards false alarms. To solve this problem, we propose a 'posterior remedy' approach. Firstly, we composed a balanced training set by duplicating some positive exemplars so that the positive and negative exemplars were equal, from which a balanced model was trained. Afterwards, the balanced model was used to predict balanced posterior probabilities, which were then remedied by a class prior probability, finally used to make decisions.

The remedy is based on the fact that the posterior probability $P(C_{hit}|d)$ is proportional to the class prior probability $P(C_{hit})$ according to the Bayesian formula,

$$P(C_{hit}|d) = \frac{p(d|C_{hit})P(C_{hit})}{p(d)} \tag{6.10}$$

Noticing that the balanced model outputs posterior probabilities assuming $P(C_{hit}) = P(C_{FA})$, we get the remedied confidence as follows:

$$P(C_{hit}|d) = \frac{\hat{P}(C_{hit}|d)P(C_{hit})}{\hat{P}(C_{hit}|d)P(C_{hit}) + (1-\hat{P}(C_{hit}|d))P(C_{FA})} \quad (6.11)$$

$$= \frac{1}{1+(1-\hat{P}(C_{hit}|d))/\hat{P}(C_{hit}|d)\ \ P(C_{FA})/P(C_{hit})} \quad (6.12)$$

where $\hat{P}(C_{hit}|d)$ is the output of the balanced model.

Note that $P(C_{hit})$ and $P(C_{FA})$ are unknown in practice, so we have to estimate them from data. According to Equation 6.11, we just need to estimate the probability ratio $P(C_{hit})/P(C_{FA})$ which can be estimated as follows:

$$P(C_{hit})/P(C_{FA}) \approx \frac{\sum_{d_i}\hat{P}(C_{hit}|d_i)}{\sum_{d_i}\hat{P}(C_{FA}|d_i)}. \quad (6.13)$$

where d_i denotes the i-th detection of the term. Obviously, the prior probability ratio is term dependent.

Besides the imbalance between positive and negative exemplars, the imbalance among different terms is also problematic. It makes the trained model biased towards frequent terms, which might affect the performance of an STD system because the evaluation metric ATWV treats frequent and infrequent terms equally. To address this problem, we balanced the training data by duplicating some exemplars of infrequent terms.

6.3.4 Discriminative decision for INV terms

We first conducted the STD experiments with INV terms applying the discriminative decision. The detailed results are reported in Table 6.1. We can see that the discriminative decision, based on either a MLP or a SVM, gave substantial performance improvement to STD. We also find that the posterior remedy refined the discriminative confidence and provided further performance improvement. A t-test shows that all the discriminative decision-based systems, either with or without the remedy, significantly outperformed the baseline system that used the lattice-based confidence ($p < 10^{-5}$). The DET curves of the various systems are shown in Figure 6.3. We can see that the discriminative decision improved the system performance with a wide range of hit/FA ratio.

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt	-	-	0.4743	0.5058
phn.ld.mlp	MLP	NO	0.5322	0.5338
phn.ld.mlp.rem	MLP	YES	**0.5453**	0.5473
phn.ld.svm	SVM	NO	0.5385	0.5413
phn.ld.svm.rem	SVM	YES	0.5432	0.5455

Table 6.1: The STD performance on INV terms based on the discriminative decision in the case of short mappings. Results based on two discriminative models, a MLP and a SVM, are reported. The best result is shown in bold face.

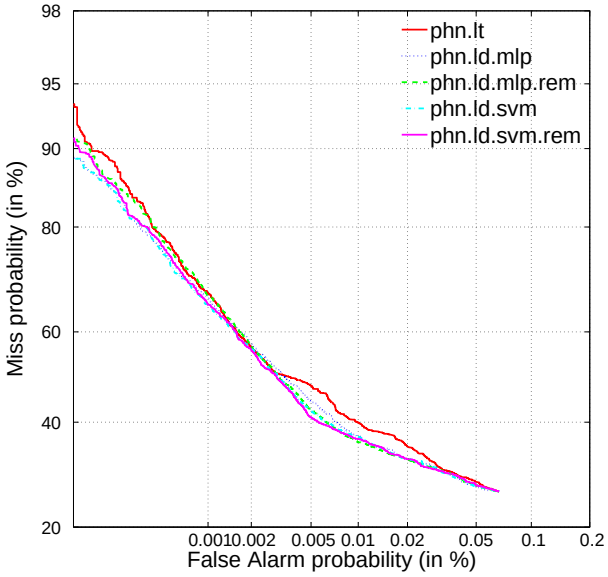

Figure 6.3: The DET curves of the STD systems based on the discriminative decision in the case of short mappings. The experiments were conducted on the INV terms.

6.3.5 Discriminative decision for OOV terms

Then we conduct experiments with the OOV terms applying the discriminative decision. The results are reported in Table 6.2, including the baseline system using the lattice-based confidence and two systems using discriminative confidence measures based on the MLP and the SVM respectively. We can see that all the systems based on discriminative confidence measures outperformed the baseline system. Another observation is that the posterior remedy did not provide further improvement, which is different from the case of INV terms. This can be attributed to the unreliable estimation

of the prior probability $P(C_{hit})$, caused by the small number of occurrences of OOV terms .

A pairwise t-test shows that the the performance improvement provided by the non-remedied discriminative is weakly significant ($p = 0.01$), whereas the improvement provided by the remedied discriminative confidence is not significant ($p \approx 0.03$). The DET curves are shown in Figure 6.4, which shows that all the discriminative decision-based systems outperformed the baseline system.

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt.jmm	-	-	0.2761	0.2770
phn.ld.mlp.jmm	MLP	NO	**0.2927**	0.2938
phn.ld.mlp.rem.jmm	MLP	YES	0.2899	0.2923
phn.ld.svm.jmm	SVM	NO	0.2894	0.2922
phn.ld.svm.rem.jmm	SVM	YES	0.2892	0.2912

Table 6.2: The STD performance on OOV terms based on the discriminative decision in the case of short mappings. Results based on two discriminative models, a MLP and a SVM, are reported. The best result is shown in bold face.

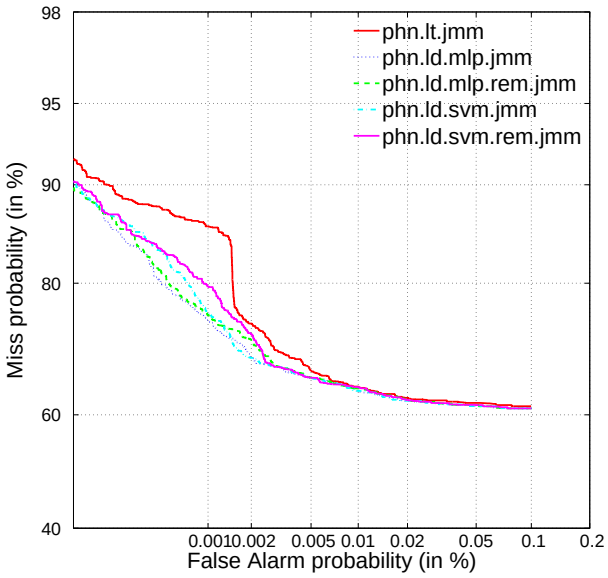

Figure 6.4: The DET curves of the STD systems based on the discriminative decision in the case of short mappings. The experiments were conducted on the OOV terms.

Comparing the MLP and SVM -based systems, we find that they exhibit similar

behaviour and provided similar performance improvement. Although the MLP-based system achieved a higher ATWV, a t-test shows that the SVM-based system exhibits a higher significance level with respect to the performance improvement, which suggests that the performance improvement achieved by the SVM-based system is more reliable.

6.3.6 Analysis of effectiveness

To gain a deeper insight for the discriminative decision, we investigate how the discriminative confidence changes the decision behaviour. We first look at the short mapping function g represented by the MLP and SVM, as shown in Figure 6.5 and 6.6 respectively. It can be seen that the MLP represents a monotonic scaling function, while the SVM represents a non-monotonic mapping function. Normally, a scaling on the confidence measure can be offset by a scaling on the decision threshold, and therefore it does not change the decision behaviour. However, when applied together with the confidence normalisation, the scaling might give substantial improvement, because the scaled confidence represents the classification posterior probability which is consistent with the confidence normalisation.

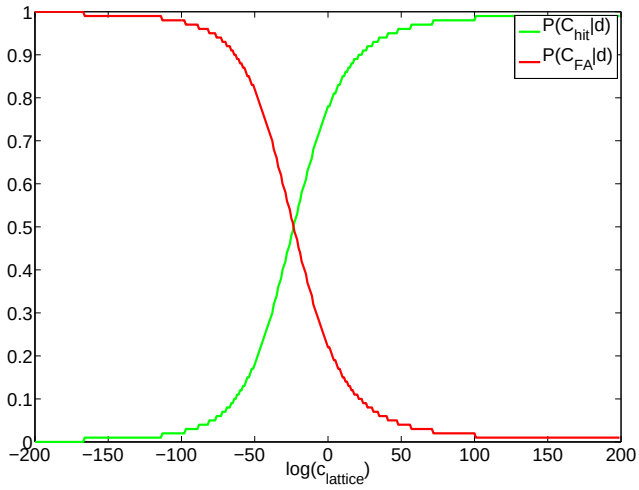

Figure 6.5: The mapping function represented by the MLP trained for the discriminative decision in the case of short mappings.

Next we examine the discriminative power of the discriminative confidence measures. For that, we plot the class-conditional confidence distributions for hits and false alarms, i.e., $p(c(d)|C_{hit})$ and $p(c(d)|C_{FA})$ respectively, and examine the overlap of

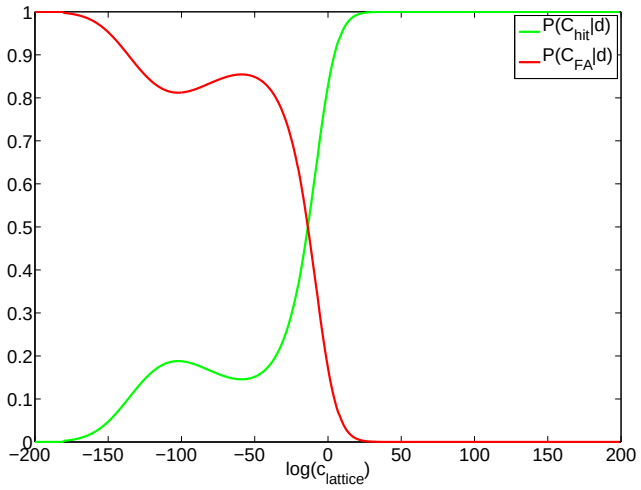

Figure 6.6: The mapping function represented by the SVM trained for the discriminative decision in the case of short mappings.

these two distributions. Obviously, a smaller overlap indicates better discrimination. In addition, to examine the effect of confidence normalisation presented in Section 5.1.3, we plot and compare the distributions of the confidence before and after normalisation. Figure 6.7 and 6.8 show the experimental results on INV terms with the MLP-based and SVM-based confidence respectively, and Figure 6.9 and 6.10 show the results on OOV terms. We can see that both the discrimination and the normalisation improved discrimination of the lattice-based confidence; when applied together, these two techniques provided more improvement than applied individually. We also see that the posterior remedy raised the discriminative power for both INV terms and OOV terms.

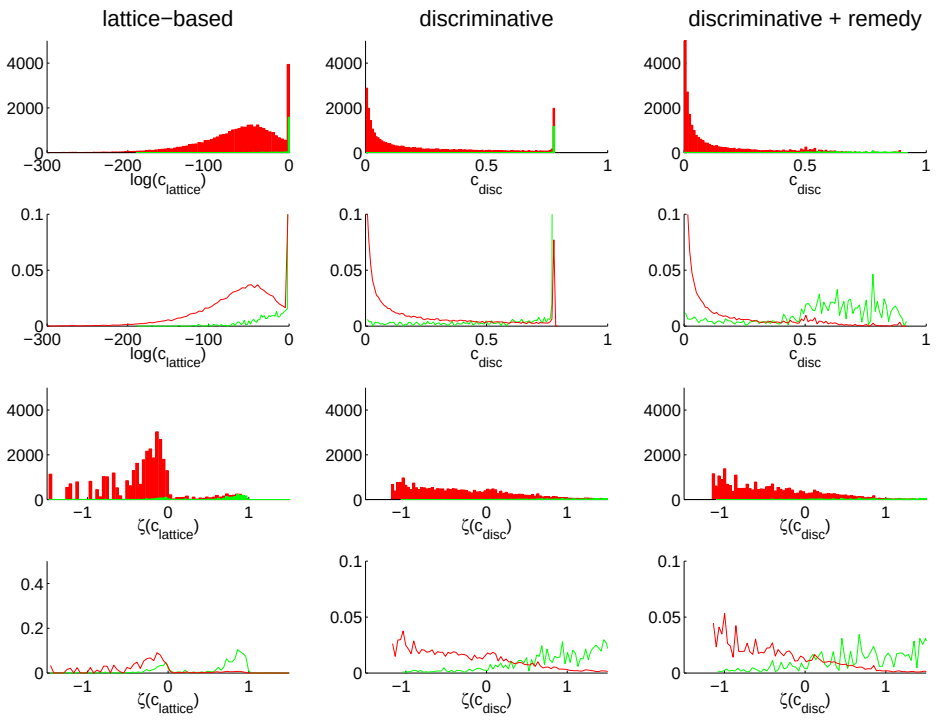

Figure 6.7: The discriminative power of the discriminative confidence measures based on the MLP, with and without posterior remedy. The detections were generated by running STD on the development set with INV terms. The plots in the first column present distributions of the lattice-based confidence. Similarly, the second column presents for the discriminative confidence, and the third column presents for the remedied discriminative confidence. In each column, the plots in the first and second row report the histogram and smoothed distribution before the normalisation, and the plots in the third and fourth row report the histogram and smoothed distribution after the normalisation. In each plot, green bars or lines represent hits, and red bars and lines represent false alarms.

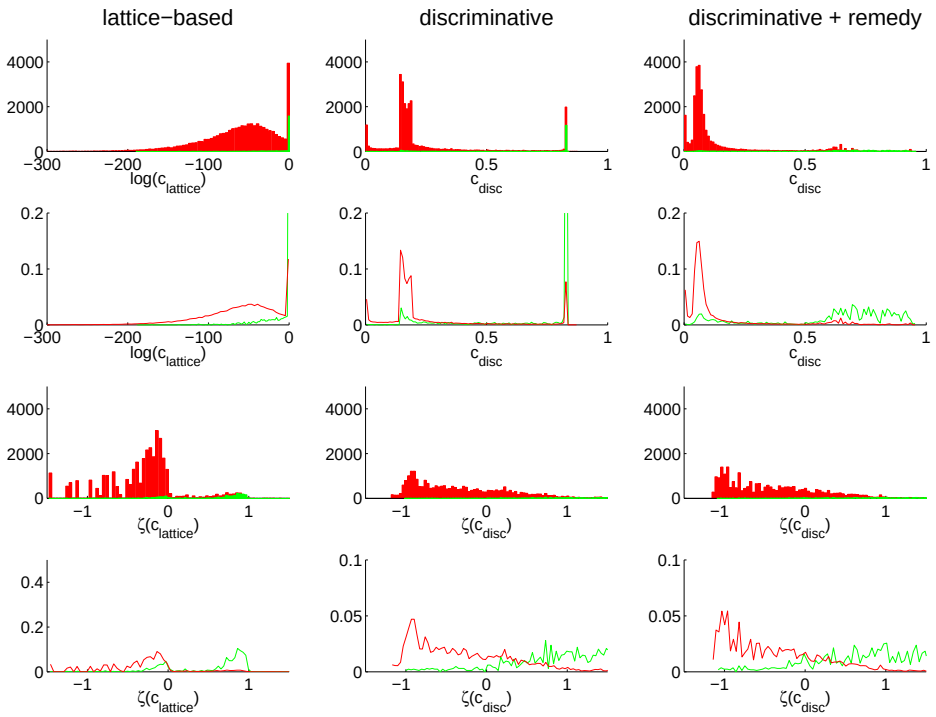

Figure 6.8: The discriminative power of the discriminative confidence measures based on the SVM, with and without posterior remedy. The detections were generated by running STD on the development set with INV terms. The meanings of the plots are the same as in Figure 6.7.

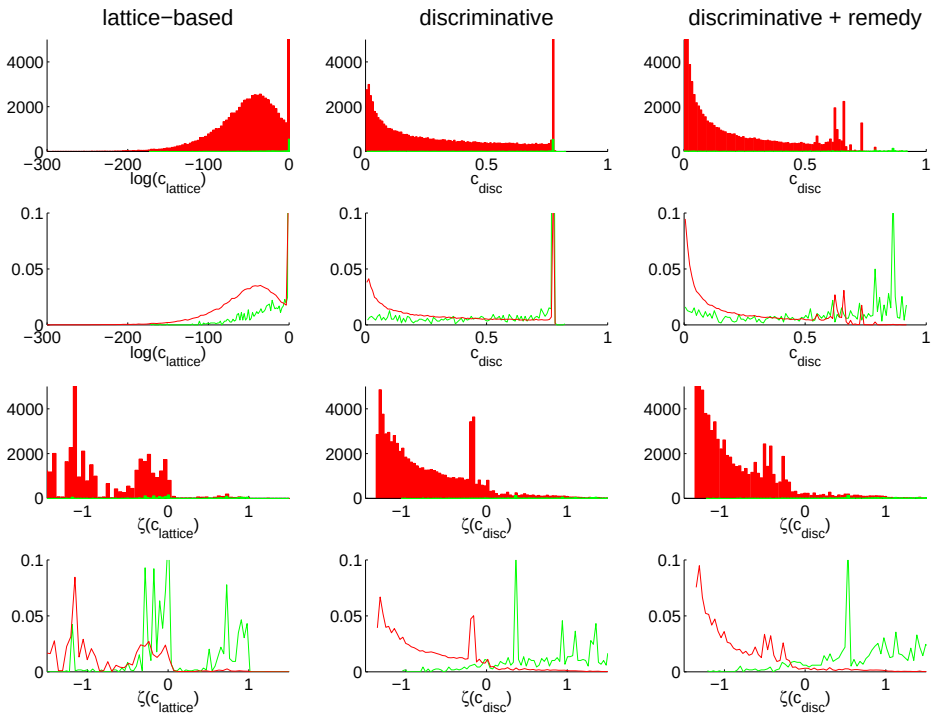

Figure 6.9: The discriminative power of the discriminative confidence measures based on the MLP, with and without posterior remedy. The detections were generated by running STD on the development set with OOV terms. The meanings of the plots are the same as in Figure 6.7.

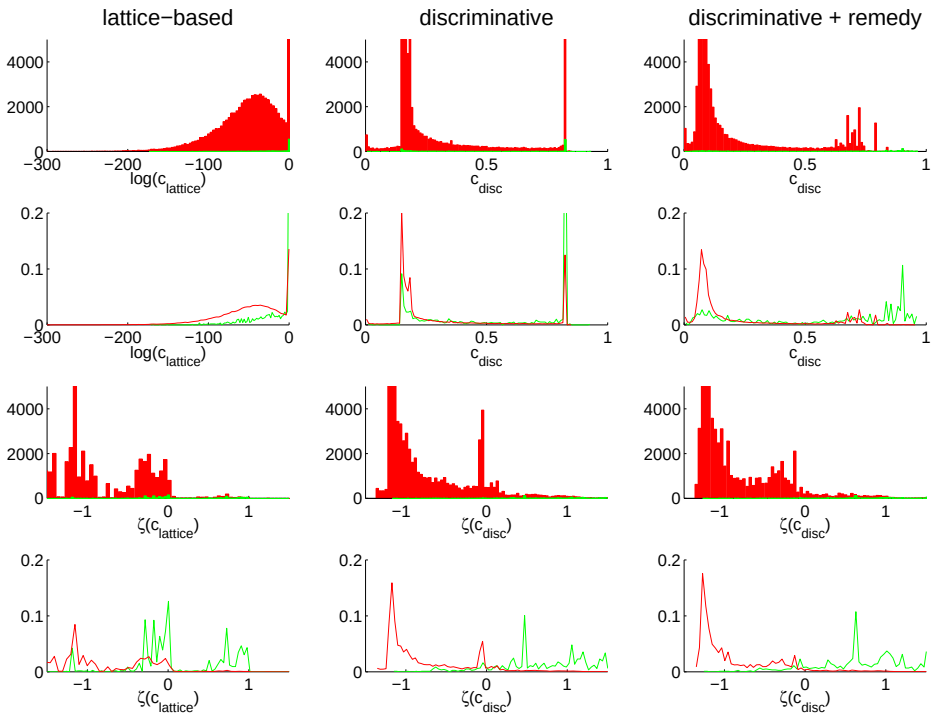

Figure 6.10: The discriminative power of the discriminative confidence measures based on the SVM, with and without posterior remedy. The detections were generated by running STD on the development set with OOV terms. The meanings of the plots are the same as in Figure 6.7.

6.4 Discriminative decision based on long mappings

Now we extend the short mapping function g to the long-mapping function f by taking into account more decision factors, which may discover some potential dependence between the discriminative confidence and the raw decision factors, especially term-dependent factors.

6.4.1 Occurrence-derived decision factors

Although any informative factor can be considered in the long mapping, term-dependent factors are preferable. We designed two occurrence-derived attributes to represent the term-dependence: effective occurrence rate $R_0(K)$, and effective false alarm rate $R_1(K)$, defined as the following:

$$R_0(K) = \frac{\sum_{d_i \in \Xi(K)} c_{lattice}(d_i)}{T} \tag{6.14}$$

and

$$R_1(K) = \frac{\sum_{d_i \in \Xi(K)} (1 - c_{lattice}(d_i))}{T} \tag{6.15}$$

where $\Xi(K)$ is the set of detections of term K, and T is the length of all the audio files. These two factors are called *occurrence-derived factors*. Note that R_0 and R_1 can not be derived from each other, because

$$R_1(K) = \frac{|\Xi(K)|}{T} - R_0(K) \tag{6.16}$$

where $|\cdot|$ denotes the size of a set. We can see that the first item on the right side of this equation is another term-dependent quantity which reflects a *detection occurrence rate* of the term K.

Applying R_0 and R_1 to the discriminative models, the discriminative mapping function is written as

$$disc(d) = f(c_{lattice}(d), R_0(K_d), R_1(K_d)). \tag{6.17}$$

where K_d is the detected term of d.

To investigate how R_0 and R_1 improve discriminative power, we conducted the STD experiment on the development set with both INV terms and OOV terms, and represent all the detections in the coordinate $c_{lat} \times R_0$ and $c_{lat} \times R_1$, as shown in Fig. 6.11, where a green cross represents a hit and a red dot represents a false alarm. In these plots, less overlap of the red dots and green crosses means more discriminative

power. It can be seen that considering the term-dependent factors can enhance the discriminative power considerably. For example, with a simple linear classifier (shown as blue lines in the plots), more false alarms can be effectively rejected than applying a threshold on the lattice-based confidence alone. Interestingly, the term-dependent factors contribute more to OOV terms than INV terms, which confirms our conjecture that considering term-dependency may compensate for the high diversity among OOV terms.

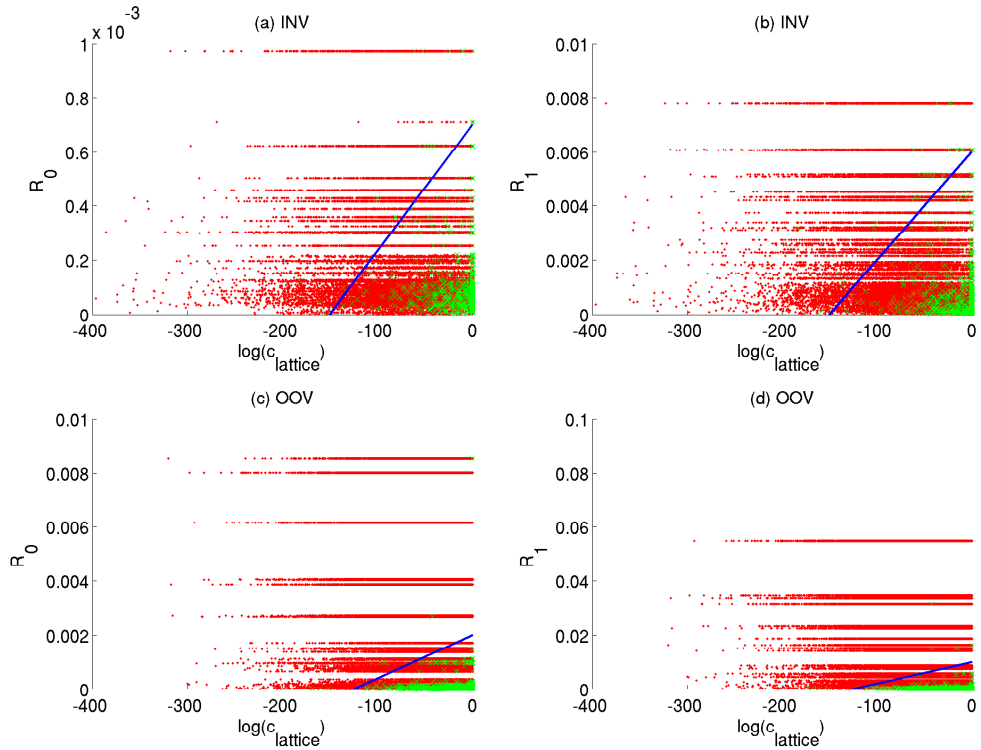

Figure 6.11: The discriminative power of R_0 and R_1. A red point represents a false alarm, and a green cross represents a hit. The blue lines represent linear hit/FA classifiers. The two plots (a)(b) show the case of INV terms, and the other two show the case of OOV terms.

Table 6.3 and 6.4 present the experimental results with R_0 and R_1 considered in the long mapping. We see that the occurrence-derived factors provided some performance improvement especially for OOV terms. This result is expected, as it can be seen from Figure 6.11 that R_0 and R_1 increase discrimination for OOV terms. A pairwise t-test shows that the performance improvement contributed by R_0 and R_1 is weakly significant ($p < 0.05$), and the discriminative decision-based systems considering R_0 and R_1 outperformed the baseline system significantly ($p < 0.01$).

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt	-	-	0.4743	0.5058
phn.ld.mlp.R	MLP	NO	0.5313	0.5334
phn.ld.mlp.R.rem	MLP	YES	**0.5460**	0.5473
phn.ld.svm.R	SVM	NO	0.5392	0.5411
phn.ld.svm.R.rem	SVM	YES	0.5421	0.5459

Table 6.3: The STD performance of discriminative decision-based systems on INV terms when considering the occurrence-derived factors R_0 and R_1. The best result is shown in bold face.

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt.jmm	-	-	0.2761	0.2770
phn.ld.mlp.R.jmm	MLP	NO	**0.2931**	0.2939
phn.ld.mlp.R.rem.jmm	MLP	YES	0.2897	0.2934
phn.ld.svm.R.jmm	SVM	NO	0.2921	0.2923
phn.ld.svm.R.rem.jmm	SVM	YES	0.2905	0.2924

Table 6.4: The STD performance of discriminative decision-based systems on OOV terms when considering the occurrence-derived factors R_0 and R_1. The best result is shown in bold face.

6.4.2 Multiple decision factors

The model-based discriminative approach allows us to take more decision factors into account to improve decision quality. In this experiment, we added a multitude of decision factors into the inputs of the discriminative models, including the acoustic likelihood (A), language model score (L) and duration (T), plus the occurrence-derived factors R_0 and R_1. The contribution of each factor can be balanced by the discriminative models implicitly. With these factors applied, the discriminative mapping is written as

$$disc(d) = f(c_{lattice}(d), A, L, T, R_0(K_d), R_1(K_d)). \tag{6.18}$$

Experimental results are shown in Table 6.5 and 6.6 for the INV terms and OOV terms respectively, and the DET curves are shown in Figure 6.12 and Figure 6.13. The results show that taking account of multiple decision factors provided marginal

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt	-	-	0.4743	0.5058
phn.ld.mlp.RTAL	MLP	NO	0.5361	0.5362
phn.ld.mlp.RTAL.rem	MLP	YES	**0.5466**	0.5467
phn.ld.svm.RTAL	SVM	NO	0.5397	0.5448
phn.ld.svm.RTAL.rem	SVM	YES	0.5434	0.5476

Table 6.5: The STD performance of the discriminative decision-based systems on INV terms when considering multiple decision factors. The inputs of the discriminative models include multiple decision factors, such as the acoustic likelihood, the language model score and the duration. The best result is shown in bold face.

System name	Model	Posterior remedy	ATWV	max-ATWV
phn.lt.jmm	-	-	0.2761	0.2770
phn.ld.mlp.RTAL.jmm	MLP	NO	**0.2952**	0.2960
phn.ld.mlp.RTAL.rem.jmm	MLP	YES	0.2918	0.2939
phn.ld.svm.RTAL.jmm	SVM	NO	0.2920	0.2936
phn.ld.svm.RTAL.rem.jmm	SVM	YES	0.2899	0.2926

Table 6.6: The STD performance of the discriminative decision-based systems on OOV terms when considering multiple decision factors. The inputs of the discriminative models include multiple decision factors, such as the acoustic likelihood, the language model score and the duration. The best result is shown in bold face.

performance improvement, although not significant statistically.

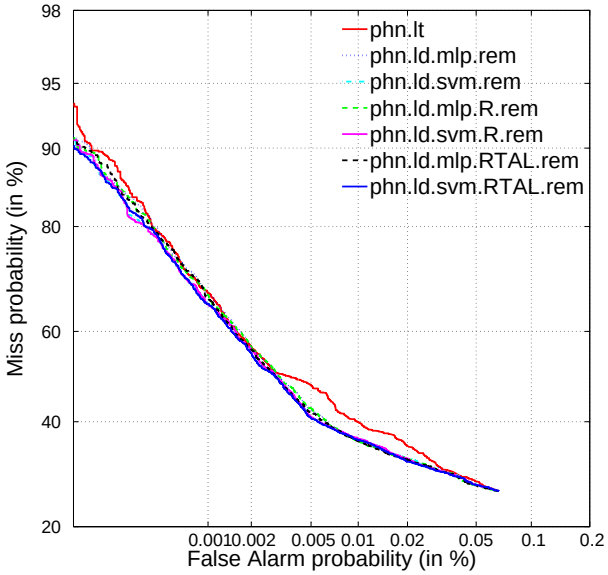

Figure 6.12: The DET curves of STD systems based on the discriminative decision in the case of long mapping. The experiments were conducted on the INV terms.

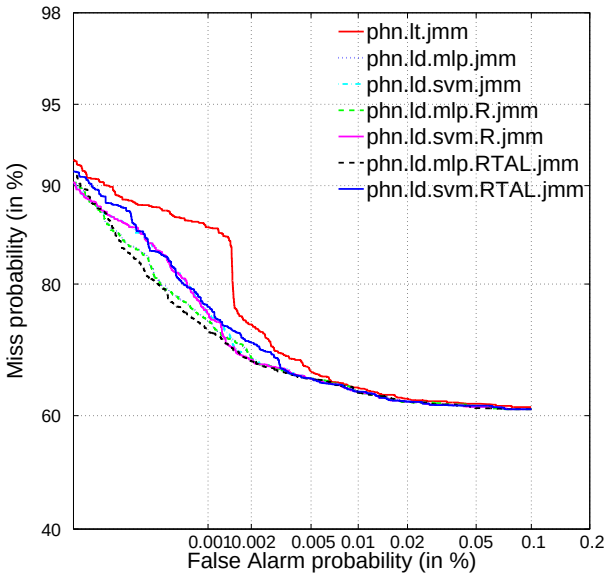

Figure 6.13: The DET curves of STD systems based on the discriminative decision in the case of long mapping. The experiments were conducted on the OOV terms.

6.5 Multiple confidence-based discriminative decision

An advantage of the model-based discriminative decision approach is that multiple confidence measures can be integrated by the discriminative model to make a composite decision. In this section, we investigate the multiple confidence-based decision based on the discriminative approach.

6.5.1 Discriminative decision for SPM

In the previous chapter, we introduced a SPM to deal with the stochastic pronunciation of OOV terms. In the SPM-based system, we used linear interpolation to combine the lattice-based confidence and the pronunciation confidence into a compound confidence to make a postponed decision. Within the framework of discriminative decision, these two confidence measures and their linear interpolation, plus other decision factors, are fed into a discriminative model, and mapped to a discriminative confidence. This can be formally written as a mapping function f:

$$c_{disc}(d) = f(c_{lattice+pron}(d), c_{lattice}(d), c_{pron}(d), R_0(K_d), R_1(K_d)) \tag{6.19}$$

where $c_{lattice}$ denotes the lattice-based confidence, c_{pron} denotes the pronunciation confidence, and

$$c_{lattice+pron} = (1-\gamma)c_{lattice}(d) + \gamma c_{pron}(d). \tag{6.20}$$

is the linear interpolation of $c_{lattice}$ and c_{pron}. R_0 and R_1 are two occurrence-derived factors. Note that the independent variables of the mapping function f in Equation 6.19 correspond to input features of discriminative models. If we do not consider model implementation, this equation can be equally written as

$$c_{disc}(d) = f(c_{lattice}(d), c_{pron}(d), \gamma, R_0(K_d), R_1(K_d)). \tag{6.21}$$

The experimental results are shown in Table 6.7. The best performance was obtained with the discriminative decision-based system whose confidence scores were predicted by the SVM model and compensated by posterior remedy. Note that the system based on the MLP-based discriminative decision outperformed the baseline system on the development set, but failed to do so on the evaluation set, indicating some over-fitting.

System name	Model	PR	ATWV(dev)	ATWV	max-ATWV
phn.lt.jmm.spm	-	-	0.2786	0.3153	0.3303
phn.ld.mlp.MPR.jmm.spm	MLP	NO	0.2979	0.2915	0.2959
phn.ld.mlp.MPR.rem.jmm.spm	MLP	YES	0.2824	0.3046	0.3321
phn.ld.svm.MPR.jmm.spm	SVM	NO	0.2747	0.3084	0.3212
phn.ld.svm.MPR.rem.jmm.spm	SVM	YES	0.2718	**0.3235**	0.3352

Table 6.7: The STD performance of the SPM-based systems on the OOV terms with discriminative confidences predicted by the MLP and the SVM. The model inputs include the lattice-based confidence and the pronunciation confidence and their interpolation, plus two occurrence-derived factors R_0 and R_1. The column 'ATWV(dev)' reports the results on the development set. 'PR' denotes 'posterior remedy'. The best result is shown in bold face.

6.5.2 Discriminative decision for soft match

Similarly, we applied the discriminative decision to soft match-based systems. A soft match-based system that allows one substitution was chosen as the baseline. The inputs of the discriminative models include the lattice-based confidence, the match confidence and their linear interpolation, as well as the occurrence-derived factors. This is formally written as follows:

$$c_{disc}(d) = f(c_{lattice+match}(d), c_{lattice}(d), c_{match}(d), R_0(K_d), R_1(K_d)) \tag{6.22}$$

where $c_{lattice}$ is the lattice-based confidence, c_{match} is the match confidence, and

$$c_{lattice+match} = (1-\nu)c_{lattice}(d) + \nu c_{match}(d). \tag{6.23}$$

is the linear interpolation of $c_{lattice}$ and c_{match}. R_0 and R_1 are two occurrence-derived factors. Again, this equation can be equally written as

$$c_{disc}(d) = f(c_{lattice}(d), c_{match}(d), \nu, R_0(K_d), R_1(K_d)). \tag{6.24}$$

The experimental results are shown in Table 6.8. We observe that the MLP-based discriminative systems outperformed the baseline system. The SVM-based discriminative systems did not outperform the baseline system, although they showed some improvement in terms of max-ATWV.

System name	Model	PR	ATWV(dev)	ATWV	max-ATWV
phn.lt.jmm.soft.n1	-	-	0.2434	0.3275	0.3300
phn.ld.mlp.MCR.jmm.soft.n1	MLP	NO	0.2748	0.3373	0.3391
phn.ld.mlp.MCR.rem.jmm.soft.n1	MLP	YES	0.2757	**0.3379**	0.3409
phn.ld.svm.MCR.jmm.soft.n1	SVM	NO	0.2611	0.3158	0.3366
phn.ld.svm.MCR.rem.jmm.soft.n1	SVM	YES	0.2419	0.3082	0.3308

Table 6.8: The STD performance of the soft match-based systems on the OOV terms with discriminative confidences predicted by the MLP and the SVM. The model inputs include the lattice-based confidence and the pronunciation confidence and their interpolation, plus two occurrence-derived factors R_0 and R_1. The column 'ATWV(dev)' reports the results on the development set. 'PR' denotes 'posterior remedy'. The best result is shown in bold face.

Comparing the results of the SPM and soft match -based system on the development set and evaluation set, we find that the results on the development set and the evaluation set are consistent with the soft match-based systems but are inconsistent with the SPM-based systems. This suggests that the discriminative models trained are more likely to be over-fitting with the SPM than with soft match.

This observation can be explained by the fact that the discriminative models are largely trained with INV terms, as discussed in Section 6.3.1. With the SPM, pronunciation expansion is rather different for INV terms and OOV terms, which causes the models trained with INV terms to fail to represent OOV terms, leading to over-fitting. With soft match, by contrast, pronunciation expansion is based on acoustic similarity and thus is not substantially different for INV terms and OOV terms, and so the models suffer less over-fitting.

Finally, we notice that the SVM-based system gave better performance than the MLP-based system with the SPM, but gave worse performance with soft match. This suggests that the MLP is good at representing the mappings whereas the SVM is good at dealing with over-fitting.

6.5.3 Discriminative decision for SPM-based soft match

Composite decision

We found in Section 5.5.3 that linear interpolation is unsuitable for combining the SPM and soft match. The failure is understandable if we notice that the two approaches produced too many inaccurate detections, such that linear interpolation is not powerful enough to describe the complex decision boundary. The multiple confidence-based decision based on the discriminative approach provides a new way to combine the SPM and soft match, by employing a discriminative model to combine the confidences from the SPM and soft match to make a composite decision.

In our experiments, three confidence measures are fed into the discriminative model, including the lattice-based confidence $c_{lattice}$, the pronunciation confidence c_{pron} and the match confidence c_{match}. In addition, the inputs also comprise the position of the pronunciation in the n-best list, denoted as I_p, and the matching level, denoted as I_m[1]. Again, this can be formally represented as a mapping function:

$$c_{disc}(d) = f(c_{lattice}(d), c_{match}(d), c_{pron}(d), I_p(d), I_m(d)) \tag{6.25}$$

The experimental results are shown in Table 6.9. We find that the SPM-based soft match systems based on discriminative models considerably outperformed the SPM-based soft match system based on linear interpolation. However, these composite systems are still much worse than the systems with the SPM or soft match individually, even with the discriminative approach applied. We conclude that the composite decision approach is not suitable for combining the SPM and soft match, even with the discriminative approach.

Detection combination

The second approach we applied to combine the SPM and soft match is the detection combination proposed in Section 5.1.4. In this approach, STD was performed separately with a system based on the SPM and a system based on soft match, and then detections from both systems were merged by a fusion technique presented in Section 5.1.4. In general, this approach is suitable for combining any heterogeneous STD systems that are complementary in detection behaviour. Herein, we combine two systems

[1]We sort and list the substitutions of a phoneme according to a confusion matrix, and assign each substitution a *substitution index* according to its position in the list. When matching a term, the substitution indexes of all matched phonemes are accumulated, giving a *matching level*.

System name	System	Model	ATWV	max-ATWV
phn.lt.jmm.spm	SPM	linear	0.3153	0.3303
phn.lt.jmm.soft.n1	soft	linear	0.3275	0.3300
phn.lt.jmm.spm.soft.n1	SPM+soft	linear	0.1790	0.2440
phn.ld.svm.MPR.rem.jmm.spm	SPM	SVM	0.3235	0.3352
phn.ld.mlp.MCR.rem.jmm.soft.n1	soft	MLP	**0.3379**	0.3409
phn.ld.mlp.IMCP.rem.jmm.spm.soft.n1	SPM+soft	MLP	0.2655	0.2839
phn.ld.svm.IMCP.rem.jmm.spm.soft.n1	SPM+soft	SVM	0.1849	0.2517

Table 6.9: The STD performance of the SPM-based soft match systems on the OOV terms based on the discriminative decision, in which 'soft' denotes 'soft match' and 'linear' denotes 'linear interpolation'.

that are based on discriminative confidence measures.

Letting d_1 and d_2 be two overlapped detections of the same term from two systems, the confidence of the merged detection can be derived as follows:

$$\begin{aligned} c &= P(C_{hit}|d_1,d_2) && (6.26)\\ &= 1-P(C_{FA}|d_1,d_2) && (6.27)\\ &= 1-P(C^1_{FA},C^2_{FA}|d_1,d_2) && (6.28)\\ &= 1-P(C^1_{FA}|d_1)P(C^2_{FA}|d_2) && (6.29)\\ &= 1-(1-P(C^1_{hit}|d_1))(1-P(C^2_{hit}|d_2)) && (6.30)\\ &= 1-(1-c_1)(1-c_2) && (6.31) \end{aligned}$$

where $c_1 = P(C^1_{hit}|d_1)$ and $c_2 = P(C^2_{hit}|d_2)$ are the confidence scores of d_1 and d_2 measured by the respective systems. In the above deduction, we have assumed that the two detection systems are independent, and a false alarm is hypothesised by the combined system only if it is hypothesised by both individual systems, i.e.,

$$C_{FA} = C^1_{FA} \wedge C^2_{FA} \tag{6.32}$$

where C_{FA} denotes a false alarm in the merged result, and C^1_{FA} and C^2_{FA} are false alarms asserted by the two systems respectively.

In practice, a scale factor α is introduced to weight the contributions of individual systems, giving Equation 6.33. We find that this is just the general form for detection

combination proposed in the previous chapter, given by Equation 5.13.

$$c = 1 - (1 - c_1)(1 - c_2)^{\alpha} \tag{6.33}$$

System name	ATWV	max-ATWV
phn.ld.svm.MPR.rem.jmm.spm	0.3235	0.3352
phn.ld.mlp.MCR.rem.jmm.soft.n1	0.3379	0.3409
phn.ld.svm.MPR.rem.jmm.spm+ phn.ld.mlp.MCR.rem.jmm.soft.n1	**0.3593**	0.3604

Table 6.10: The STD performance on the OOV terms of the combined system that merges detections from a SPM-based system and a soft match-based system, based on discriminative confidence measures. *phn.ld.svm.MPR.rem.jmm.spm* is the best SPM-based system, *phn.ld.mlp.MCR.rem.jmm.soft.n1* is the best soft match-based system, and *phn.ld.svm.MPR.rem.jmm.spm+phn.ld.mlp.MCR.rem.jmm.soft.n1* is the combined system. The best result is shown in bold face.

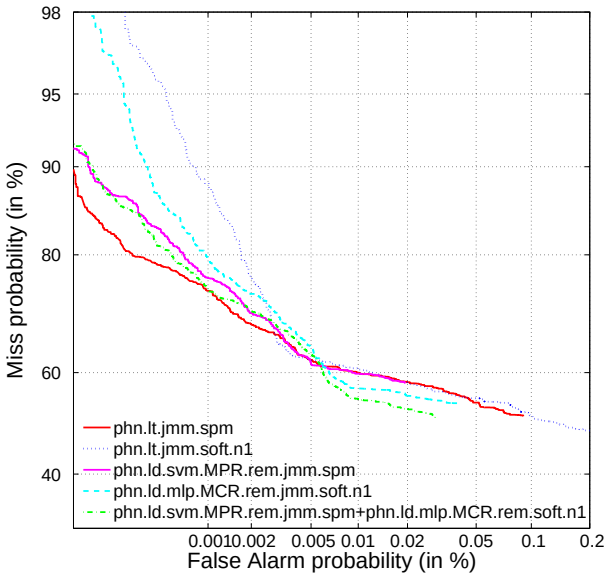

Figure 6.14: The DET curves of the SPM and soft match -based systems, as well as their detection combination based on discriminative confidence measures. The experiments were conducted on the OOV terms.

This fusion technique was applied to combine the best SPM-based system and the best soft match-based system. The results are reported in Table 6.10. We can see that

the combination provided substantial performance improvement over individual systems. A t-test shows that the improvement over both individual systems is significant ($p < 0.01$).

Figure 6.14 shows the DET curves of the lattice-based systems and the discriminative decision-based systems, when applying the SPM and soft match. The combined system is reported as well. We can see that the discriminative decision provided systematic performance improvement for the soft match-based system, but failed to do so for the SPM-based system. The combined system outperformed individual systems if a high false alarm rate is allowed, which is a normal observation with the detection combination.

6.6 Summary

In this chapter, we investigated the discriminative decision strategy. We found that the discriminative decision, based on either a MLP or a SVM, provided significant performance improvement for both INV and OOV terms. Furthermore, systems based on both the SPM and soft match for stochastic pronunciation treatment were improved by the discriminative decision. Finally, merging the SPM and soft match -based systems by detection combination gave significant performance improvement.

We acknowledge that the idea of the model-based discriminative confidence was initially presented for utterance verification, based on a MLP [Mathan and Miclet, 1991] or a SVM [Zhang and Rudnicky, 2001]. We also note that a MLP-based confidence estimation has been presented by SRI&OGI for STD [Vergyri et al., 2007]. The contribution of our study is that we extend the MLP or SVM -based confidence to a general idea of discriminative decision for STD, and prove its consistency with the confidence normalisation. Moreover, we focus on OOV terms in particular.

A potential problem, for both the lattice-based and the discriminative confidence, is that these confidence measures are largely derived from the acoustic and language models. However, these models are usually biased towards INV terms and represent OOV terms less well, which makes the confidence highly unreliable for OOV term detection. In the next chapter, we propose a novel direct posterior confidence estimation which is based on a MLP to estimate the acoustic confidence so that the weak modelling on OOV terms can be alleviated.

Chapter 7

Direct posterior confidence

In previous chapters, we have proposed a stochastic pronunciation model to handle the uncertainty in pronunciations of OOV terms, and a discriminative decision strategy to improve the detection quality. However, all these techniques work with a confidence that is derived from some generative models, for example, acoustic likelihood estimated by HMM-based acoustic models and LM scores estimated by n-gram language models. This *generative confidence* is not reliable for OOV term detection, as the generative models trained on INV terms tend to model OOV terms weakly.

In this chapter, we propose a direct posterior confidence which, based on posterior probabilities estimated with a discriminative model, is a *discriminative confidence*. This new confidence measure provides more discriminative power for decision making and alleviates the weak modelling on OOV terms with a generative confidence.

In the following sections, we first present the general idea of this new confidence estimation, and then apply it to detect INV and OOV terms respectively; finally we apply this confidence to systems based on stochastic pronunciation models and discriminative decisions.

7.1 MLP-based posterior confidence

It is well known that a standard 3-layer MLP network with softmax output activation can be used to estimate class posterior probabilities for a classification task. MLPs have been widely used in this fashion for speech recognition, by estimating the posterior probabilities for phone classes, given acoustic features as input [Hermansky et al., 2000]. Here, we use an MLP to estimate the posterior probability $P(Q_t|O)$ for each frame t, where Q_t is the phone class of the search term K at frame t, and is obtained

from the subword unit lattice produced by the recogniser. The structure of the MLP is shown in Figure 7.1.

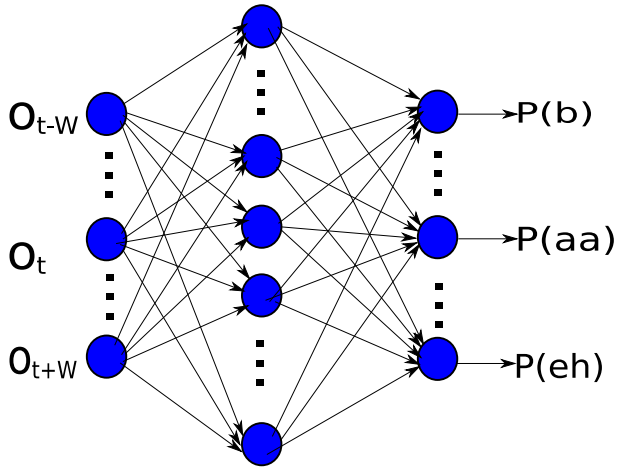

Figure 7.1: The MLP structure for framewise posterior probability estimation. The input layer consists of 9 frames of 39-dimension PLP features, amounting to 351 input nodes, and the outputs are phone categories, which for English include 40 vowels and consonants plus a short and a long silence. The hidden layer, whose size is optimised by cross-validation, contains 5k hidden units.

Now consider the confidence of a detection d defined as follows,

$$d = (K, s = (t_1, t_2), v_a, v_l, ...) \tag{7.1}$$

where K is the detected term, s is the speech segment from t_1 to t_2 within which the detection resides, and v_a and v_l are the acoustic score and LM score respectively. With the framewise posterior probability $P(q_t|O)$, the confidence of d is calculated simply by summing the frame confidences, as shown in Equations 7.2-7.4.

$$\begin{aligned} c_{mlp}(d) &= P(K_{t_1}^{t_2}|O) && (7.2)\\ &= \prod_{t=t_1}^{t_2} P(q_t|O) && (7.3)\\ &= \prod_{t=t_1}^{t_2} P(q_t|o_{t-W}, ..., o_t, ...o_{t+W}) && (7.4) \end{aligned}$$

We assume here that the confidence of each frame is independent, and both phones and search terms are independent as well. This assumption makes the MLP-based posterior probability a *local* confidence. Figure 7.2 shows the graphical representation of this phone-independent approach.

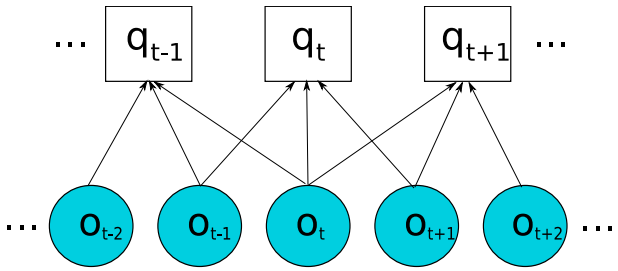

Figure 7.2: The graphical representation of the phones-are-independent model for posterior confidence calculation. q_t is the phone class at frame t, and o_t is the observed acoustic feature at time t.

We call the MLP-based confidence a *direct posterior confidence* since it is based on posterior probabilities calculated from a discriminative model (MLP here) directly, instead of resorting to the Bayesian formula as the lattice-based confidence estimation does. Note that the direct posterior confidence is not necessarily based on a MLP, but any discriminative model that evaluates the posterior probability locally.

7.1.1 LM posterior confidence

The strong phones-are-independent assumption above leads to a simple local confidence measure, but it also means that some useful information from linguistic constraints is ignored. To remedy this, dependence should be added between phones, as shown in Figure 7.3.

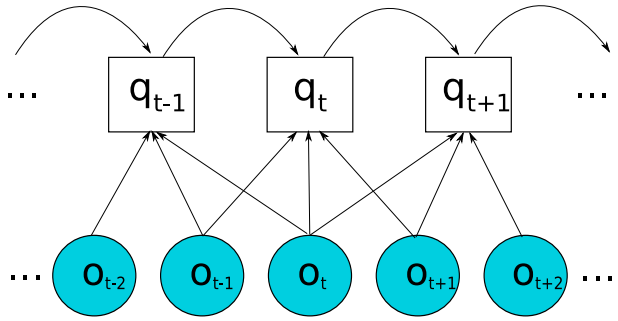

Figure 7.3: The graphical representation of the phones-are-dependent model for posterior confidence calculation. q_t is the phone at frame t, and o_t is the observed acoustic feature at time t. In this model, the phone dependency is described by a 2-gram LM.

Note that the introduced dependence is among phones, and hence is a linguistic constraint. The evidence of a putative detection provided by the linguistic constraint can be represented by a posterior probability of the phoneme string of the search term

given the lattice. This can be formulated as Equation 7.5-7.7, where L denotes the entire phoneme lattice, K^l denotes the phoneme form of K, and C_{K^l} is the context of K^l.

$$\begin{aligned} c_{lm}(d) &= P(K^l|L) && (7.5)\\ &= \frac{P(K^l,L)}{P(L)} && (7.6)\\ &= \frac{\sum_{C_{K^l}} P(K^l,C_{K^l})}{P(L)} && (7.7)\\ & && (7.8) \end{aligned}$$

Since the posterior probability $P(K^l|L)$ concerns linguistic constraints only, we denote c_{lm} as a *LM posterior confidence*; correspondingly, the direct posterior confidence c_{mlp} is an *acoustic posterior confidence*.

7.1.2 Confidence integration

The acoustic and LM posterior confidence describe different aspects of a detection, thus can be combined to give a better decision. We tested several approaches, and found that the confidence integration that has been presented for detection combination gave the best result, formulated as follows:

$$\begin{aligned} c_{mlp+lm} &= 1-(1-c_{mlp})^{\alpha}(1-c_{lm}) && (7.9)\\ &= 1-(1-P(K_{t_1}^{t_2}|O))^{\alpha}(1-P(K^l|K)) && (7.10) \end{aligned}$$

where α is a scale factor, and c_{mlp+lm} is the integrated confidence.

This integration approach can be extended to combine the direct posterior confidence and the lattice-based confidence, which gives rise to

$$\begin{aligned} c_{mlp+lattice} &= 1-(1-c_{mlp})^{\alpha}(1-c_{lattice}) && (7.11)\\ &= 1-(1-P^{mlp}(K_{t_1}^{t_2}|O))^{\alpha}(1-P^{lat}\ (K_{t_1}^{t_2}|O)) && (7.12) \end{aligned}$$

where $c_{mlp+lattice}$ is the integrated confidence, $P^{mlp}(K_{t_1}^{t_2}|O)$ is calculated from the MLP and $P^{lat}(K_{t_1}^{t_2}|O)$ is calculated from the lattice.

System name	Term	Confidence	Post. conf.	ATWV	max-ATWV
phn.lt	INV	$c_{lattice}$	NO	0.4743	0.5058
phn.pt	INV	c_{mlp}	YES	0.4902	0.4994
phn.pt.c1	INV	c_{mlp+lm}	YES	0.4963	0.5022
phn.pt.c2	INV	$c_{mlp+lattice}$	YES	**0.5344**	0.5363
phn.lt.jmm	OOV	$c_{lattice}$	NO	0.2761	0.2770
phn.pt.jmm	OOV	c_{mlp}	YES	0.2971	0.2986
phn.pt.c1.jmm	OOV	c_{mlp+lm}	YES	0.2941	0.2980
phn.pt.c2.jmm	OOV	$c_{mlp+lattice}$	YES	**0.2973**	0.3011

Table 7.1: The performance of STD systems on INV and OOV terms when utilising the direct posterior confidence. $c_{lattice}$ denotes the lattice-based confidence, and c_{mlp} denotes the direct posterior confidence. c_{mlp+lm} and $c_{mlp+lattice}$ are two integrated confidences presented by Equation 7.9 and Equation 7.11 respectively. 'Post. conf.' specifies if the system uses a direct posterior confidence.The best results on INV terms and OOV terms are shown in bold face.

7.1.3 Experimental results

In this section, we apply the direct posterior confidence to STD with basic configurations, i.e., neither stochastic pronunciations nor soft match is used. The results are presented in Table 7.1, and the DET curves are shown in Figure 7.4 and Figure 7.5 for INV and OOV terms respectively.

From the results above, we can see that for both INV terms and OOV terms, systems based on the direct posterior confidence c_{mlp} considerably outperformed the baseline system that uses the lattice-based confidence. Meanwhile, we find that the behaviour of the systems with the direct posterior confidence is different with INV terms and OOV terms. With OOV terms, the direct posterior confidence c_{mlp} performed significantly better than the lattice-based confidence $c_{lattice}$ ($p < 0.01$), but provided no further improvement when integrated with the LM posterior confidence c_{lm}. By contrast, c_{mlp} did not provide any significant improvement for the INV terms ($p = 0.2$), but gave rather significant performance improvement when integrated with c_{lm} ($p < 1e-5$).

These observations provide strong evidence for our conjecture that OOV terms are badly modelled by acoustic and language models and thus the lattice-based confidence

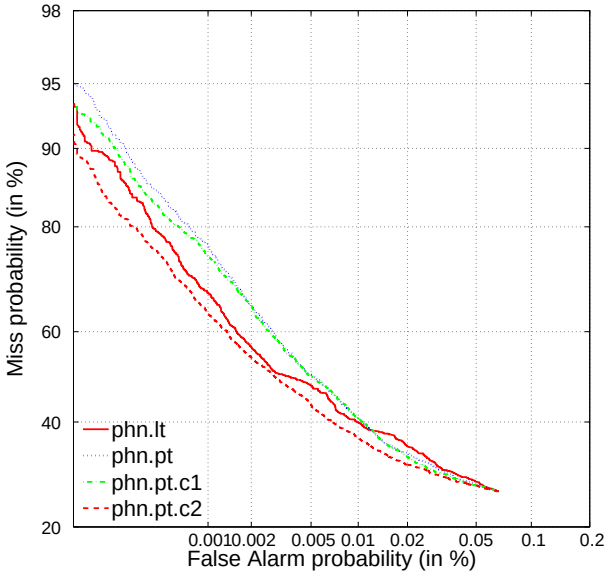

Figure 7.4: The DET curves of the STD systems using the direct posterior confidence. The experiments were conducted with INV terms.

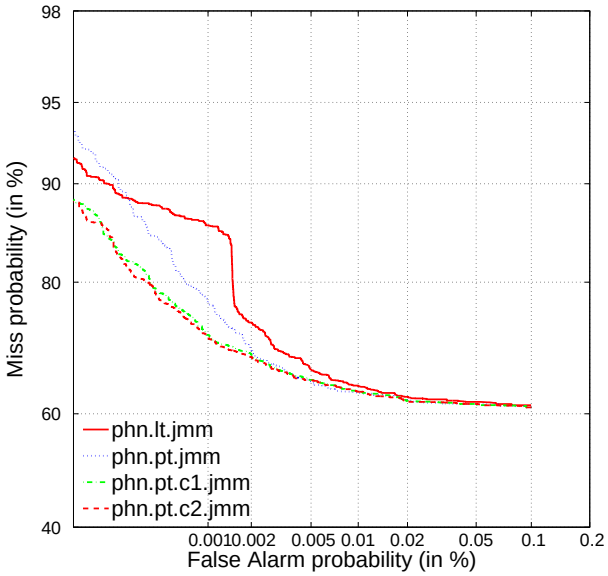

Figure 7.5: The DET curves of the STD systems using the direct posterior confidence. The experiments were conducted with OOV terms.

tends to be inaccurate. For that reason, the direct posterior confidence is highly desirable for OOV term detection.

Finally, we notice that the integrated confidence $c_{mlp+lattice}$ provided the best per-

formance for both INV and OOV terms, though more substantially for INV terms. This is because the direct posterior confidence and the lattice-based confidence are derived from different models that are based on different rationalities, thus are complementary.

7.2 Direct posterior confidence with SPM and soft match

In this experiment, we apply the direct posterior confidence to the SPM and soft match -based systems for OOV term detection. The idea is to combine the power of the direct posterior confidence in discriminating competing pronunciations and the power of the SPM and soft match in dealing with pronunciation variation. A linear interpolation is used to integrate multiple confidences, as formulated in Equation 7.13 and 7.14 for the SPM and soft match -based systems respectively. Note that these formulas are similar to Equation 5.24 and 5.34, except that the direct posterior confidence c_{mlp} replaces the lattice-based confidence $c_{lattice}$.

$$c_{mlp+pron}(d) = (1-\gamma)c_{mlp}(d) + \gamma c_{pron}(d) \tag{7.13}$$

$$c_{mlp+match}(d) = (1-\nu)c_{mlp}(d) + \nu c_{match}(d) \tag{7.14}$$

where γ and ν are scale factors.

The experimental results are reported in Table 7.2. We can see that the direct posterior confidence improved both the SPM-based system and the soft match-based system. A t-test shows that the performance improvement provided by the direct posterior confidence is weakly significant ($p \approx 0.06$) for the SPM-based system, though not significant for the soft match-based system. The DET curves are shown in Figure 7.6, which further confirms that systems based on the direct posterior confidence systematically outperform systems based on the lattice-based confidence. We also find that with the direct posterior confidence, the SPM-based system exhibited much superiority over the soft match-based system in most of the operating region, as we have seen in the experiments with the lattice-based confidence.

System name	Pron. var.	Post. conf.	ATWV	max-ATWV
phn.lt.jmm	NONE	NO	0.2761	0.2770
phn.lt.jmm.spm	SPM	NO	0.3153	0.3303
phn.lt.jmm.soft.n1	soft match	NO	0.3275	0.3300
phn.pt.jmm	NONE	YES	0.2971	0.2986
phn.pt.jmm.spm	SPM	YES	0.3288	0.3332
phn.pt.jmm.soft.n1	soft match	YES	**0.3387**	0.3505

Table 7.2: The performance of STD systems on OOV terms when applying the direct posterior confidence together with the SPM or soft match. The systems with the lattice-based confidence are also reported for comparison. 'Pron. var.' specifies how pronunciation variation is treated, and 'Post. conf.' specifies if the system uses a direct posterior confidence.

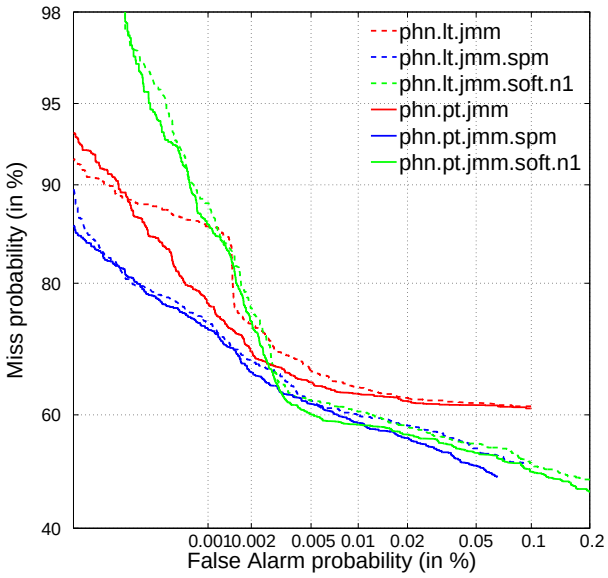

Figure 7.6: The DET curves of the STD systems applying the direct posterior confidence along with the SPM and soft match. The experiments were conducted with OOV terms.

7.3 Direct posterior confidence with discriminative decision

We have shown in Chapter 6 that the discriminative decision strategy can improve the performance of an STD system significantly. In this section, we combine the direct posterior confidence estimation and the discriminative decision. A simple way is to treat the direct posterior confidence as one of the decision factors that are fed into discriminative models for discriminative confidence estimation.

7.3.1 Direct posterior confidence with discriminative decision for INV terms

We first examine the performance of systems applying the direct posterior confidence together with the discriminative decision when detecting INV terms. In this case, the discriminative model can be written as a mapping function f as follows,

$$c_{disc}(d) = f(c_{mlp+lattice}(d), c_{lattice}(d), c_{mlp}(d), R_0(K_d), R_1(K_d)) \tag{7.15}$$

where R_0 and R_1 are the occurrence-derived factors.

In this experiment, we chose a MLP as the discriminative model since it has given better performance than a SVM for INV term detection in Section 6.4. Table 7.3 reports the experimental results. We see that the direct posterior confidence and the discriminative decision improved system performance significantly ($p < 0.01$); however, combining these two techniques did not give further performance increase. This can be attributed to the fact that both these two techniques attempt to enhance the discriminative power of the confidence measure, therefore they are not very complementary.

The DET curves are shown in Figure 7.7 , which shows again that the direct posterior confidence and the discriminative decision individually improved the system performance, whereas combining them did not provide further improvement.

7.3.2 Direct posterior confidence with discriminative decision for OOV terms

It is more complex when we apply the direct posterior confidence together with the discriminative decision to OOV term detection, since we need to examine systems with different ways of treating pronunciation variation.

System name	Post. conf.	Disc. dec.	ATWV	max-ATWV
phn.lt	NO	NONE	0.4743	0.5058
phn.ld.mlp.R.rem	NO	MLP	**0.5460**	0.5473
phn.pt	YES	NONE	0.4902	0.4994
phn.pd.mlp.MVR.rem	YES	MLP	0.5391	0.5470

Table 7.3: The performance of the phoneme-based STD systems on INV terms when applying the direct posterior confidence and the MLP-based discriminative decision. 'Post. conf.' specifies if the system uses a direct posterior confidence, and 'Disc. dec.' specifies which model the discriminative decision is based on, if applied. The best result is shown in bold face.

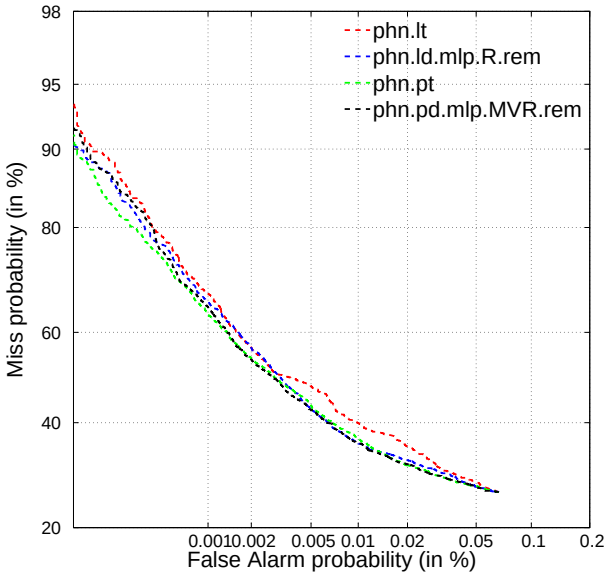

Figure 7.7: The DET curves of STD systems applying direct posterior confidence and discriminative decision making. The experiments were conducted with INV terms.

For the SPM-based system, we chose an SVM as the discriminative model because it exhibited better performance in systems with the lattice-based confidence (Section 6.5.1). The mapping function can be written as

$$c_{disc}(d) = f(c_{mlp+lattice}(d), c_{lattice}(d), c_{mlp}(d), c_{pron}(d), R_0(K_d), R_1(K_d)) \qquad (7.16)$$

where c_{pron} is the pronunciation confidence.

For the soft match-based system, we chose an MLP as the discriminative model as

it exhibited better performance in systems with the lattice-based confidence (Section 6.5.2). The mapping function can be written as

$$c_{disc}(d) = f(c_{mlp+lattice}(d), c_{lattice}(d), c_{mlp}(d), c_{match}(d), R_0(K_d), R_1(K_d)) \quad (7.17)$$

where c_{match} is the match confidence.

Table 7.4 shows the experimental results. An interesting observation is that combining the direct posterior confidence and the discriminative decision gave better performance than applying individual techniques. This suggests that for OOV terms, the weak discrimination needs to be treated by both confidence estimation and decision making.

Finally, we combined the best SPM-based system and the best soft match-based system by detection combination. The result, as shown in Table 7.4, is better than either individual system. In fact, this is the best result we obtained with the OOV terms.

The DET curves are shown in Figure 7.8 for various systems based on the direct posterior confidence and the discriminative decision. We can see again that the direct posterior confidence plus the discriminative decision gave better performance than applying individual techniques.

7.4 Summary

In this chapter, we presented a novel direct posterior confidence for STD, especially for OOV term detection. Experimental results confirmed that this new confidence can improve the performance of an STD system significantly with OOV terms. It also performed well with various approaches used to treat pronunciation uncertainty, such as SPM and soft match, and provides additional performance improvement when applied together with the discriminative decision.

System name	Pron. var.	Post. conf.	Disc. dec.	ATWV	max-ATWV
phn.lt.jmm.spm	SPM	NO	NONE	0.3153	0.3303
phn.pt.jmm.spm	SPM	YES	NONE	0.3288	0.3332
phn.ld.svm.MPR.rem.jmm.spm	SPM	NO	SVM	0.3235	0.3352
phn.pd.svm.MVPR.rem.jmm.spm	SPM	YES	SVM	0.3318	0.3502
phn.lt.jmm.soft.n1	soft	NO	NONE	0.3275	0.3300
phn.pt.jmm.soft.n1	soft	YES	NONE	0.3387	0.3505
phn.ld.mlp.MCR.rem.jmm.soft.n1	soft	NO	MLP	0.3275	0.3300
phn.pd.mlp.MVCR.rem.jmm.soft.n1	soft	YES	MLP	0.3516	0.3571
phn.ld.svm.MPR.rem.jmm.spm+ phn.ld.mlp.MCR.rem.jmm.soft.n1	soft+SPM	No	SVM+MLP	0.3593	0.3604
phn.pd.svm.MVPR.rem.jmm.spm+ phn.pd.mlp.MVCR.rem.jmm.soft.n1	soft+SPM	YES	SVM+MLP	**0.3692**	0.3728

Table 7.4: The performance of the STD systems on OOV terms when applying the direct posterior confidence and the discriminative decision. 'Pron. var.' specifies the approach to deal with pronunciation variation; 'soft' denotes 'soft match'; 'Post. conf.' specifies if the system uses a direct posterior confidence, and 'Disc. dec.' specifies which model the discriminative decision is based on, if applied. The best result is shown in bold face.

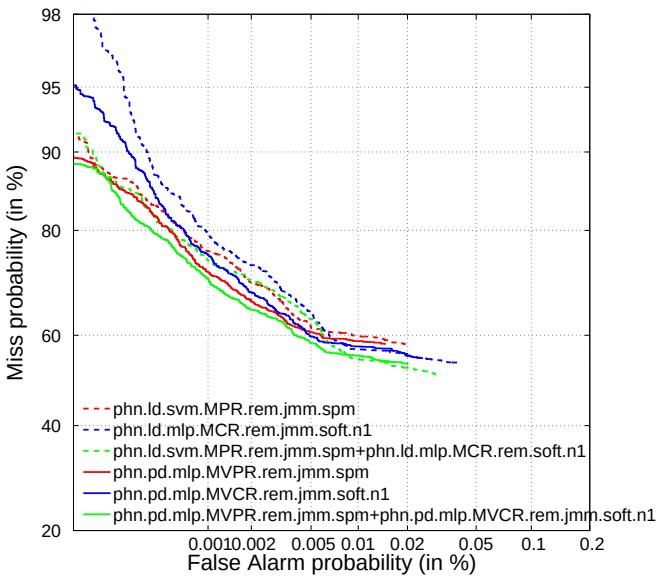

Figure 7.8: The DET curves of STD systems applying posterior confidence and discriminative decision making when detecting OOV terms. The systems based on the lattice-based confidence are also reported for comparison.

Chapter 8

Conclusion

8.1 Contributions

In this thesis, we focused on improving the performance of an STD system on OOV terms. We hypothesised that the challenges in detecting OOV terms are caused by the particular properties of OOV terms and that techniques which address these properties will improve OOV detection. With a series of studies and experiments presented in previous chapters, we can now conclude that this hypothesis has been proved.

In summary, the contributions of this thesis comprise three aspects, targeting three challenges of OOV terms: pronunciation uncertainty, property diversity and weak modelling.

Stochastic pronunciation model The first contribution of this thesis is a stochastic pronunciation model (SPM) based on joint-multigrams. This model, by considering multiple pronunciations during term detection, simulates the phonetic variation when unknown words are spoken, thus compensating for the uncertainty in OOV pronunciations. Our experiments confirmed that the SPM-based approach significantly improved the STD performance on OOV terms. Compared with the conventional soft match-based approach, the SPM-based approach exhibited better performance, especially when the false alarm rate was low.

Term-dependent discriminative decision The second contribution of this work is that we proposed a term-dependent discriminative decision strategy to deal with the idiosyncrasy and diversity of OOV terms. We applied discriminative models (MLP and SVM) to integrate the decision factors into a classification posterior probability,

which is discriminative and consistent with the ATWV-oriented confidence normalisation. The hit/FA decision based on this discriminative confidence measure is term-dependent, and yields minimum decision errors. Experimental results demonstrated that this decision strategy significantly improved the STD performance on OOV terms.

Direct posterior confidence The third contribution of the work is a direct posterior confidence measure. Specifically, we proposed to use a discriminative model (MLP) to compute the acoustic confidence of an OOV detection, so that we avoid using the acoustic and language models that tend to model OOV terms weakly. In our experiments, the new confidence gave better performance than the conventional lattice-based confidence on OOV terms, and improved the performance further when combined with the LM posterior probability and the lattice-based confidence. Finally, we applied the new confidence together with SPM and discriminative decision making, and achieved further performance improvement.

Although these techniques were originally proposed to enhance OOV term detection, they improved the performance on INV terms as well. As a summary, the contributions of these techniques are illustrated in Figure 8.1, where the results of word and phoneme -based systems on INV and OOV terms are reported respectively.

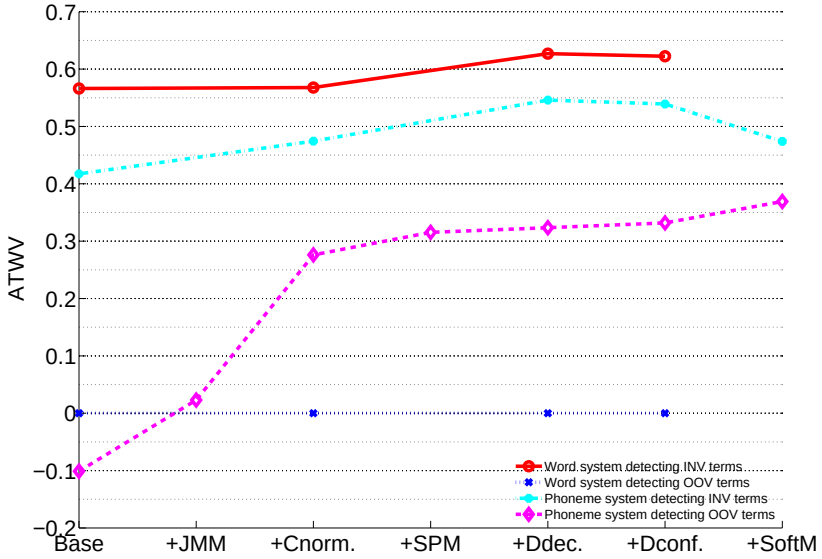

Figure 8.1: The contributions of the techniques proposed in this thesis. 'Base' is the baseline system; 'JMM' denotes the joint-multigram model-based pronunciation prediction; 'Cnorm' is the confidence normalisation; 'SPM' is the SPM-based pronunciation variation compensation; 'Ddec.' is the discriminative decision; 'Dconf.' is the direct posterior confidence estimation; '+SoftM' denotes the detection combination with the soft match-based system. As the word-based system can not detect any OOV terms, it always obtains zero ATWV in OOV detection. Note that the soft match approach does not help the phoneme-based system on INV terms, which can be attributed to the multiple pronunciation dictionary used in our experiments.

8.2 Application in practice

8.2.1 Usability

In this section, we discuss the usability of the proposed techniques in practice. First of all, we examine how much benefit our systems can provide to end users. For that purpose, we compare the performance of our best systems with the performance of random spotting. Figure 8.2 shows the DET curves on INV terms and Figure 8.3 shows those on OOV terms. We can clearly see that using our systems, users have much more chance to catch occurrences of the search terms, compared to 'blind listening' to the audio, especially when searching for OOV terms.

It is difficult to identify a performance level with which an STD system can be said to be 'usable'; however, applications based on the state-of-the-art technologies are indeed becoming commercial reality. For example, Google has released their 'audio indexing' in July, 2008. The results reported in this thesis are comparable to the state-of-the-art, and the performance gain observed in our experiments is expected to be migrated to other systems with the proposed techniques applied. In that sense, we have reasons to anticipate that the work reported in this thesis will contribute to the practical usage of speech technology.

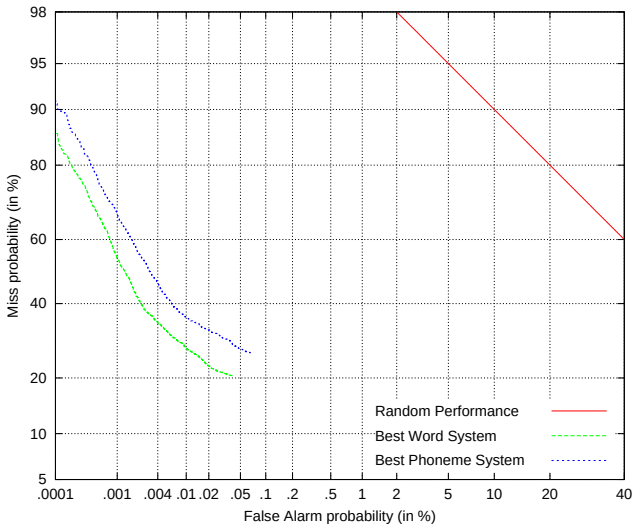

Figure 8.2: The DET curves of our best word and phoneme -based STD systems and random spotting on INV terms.

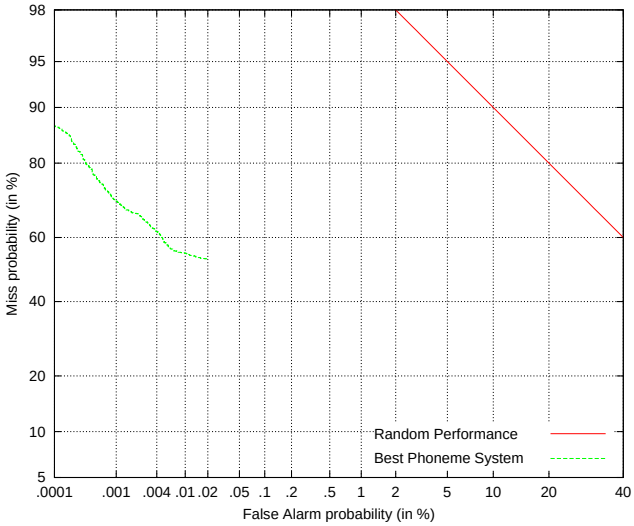

Figure 8.3: The DET curves of our best phoneme-based STD systems and random spotting on OOV terms.

8.2.2 Computational efficiency

Another concern regarding the usability of the proposed techniques is their efficiency in computation. In Section 5.5.2 we have seen that the SPM-based approach caused just insignificant computation increase (15% in our investigation experiment). For the discriminative decision making, the MLP and SVM models are very simple, and the additional computation can be ignored. For the direct posterior confidence estimation, the MLP-based posterior probabilities can be pre-computed and saved in lattices, which makes the online computation cost ignorable as well.

8.2.3 The best system

In real application development, the primary goal is to obtain the best performance by assembling all available techniques. It is well known that word-based systems outperform phoneme-based systems when detecting INV terms, while phoneme-based systems are indispensable for OOV term detection. Therefore, a commonly used approach to boost entire STD performance is to combine these two types of systems. In such a combined system, a phoneme-based system is responsible for OOV term detection only, leaving INV terms to a word-based system.

Given the performance of a STD system on INV and OOV terms, the overall performance of this system is calculated as Equation 8.1, where $ATWV_{inv}$ and $ATWV_{oov}$

denote the performance on INV and OOV terms respectively, and κ is the OOV rate.

$$ATWV_{overall} = (1-\kappa) \times ATWV_{inv} + \kappa \times ATWV_{oov} \tag{8.1}$$

Figure 8.4 shows the overall performance of the word and phoneme -based system, as well as their combination. It shows that whenever the OOV rate exceeds 18%, the phoneme-based system will outperform the word-based system; however the combined system always outperforms any individual system.

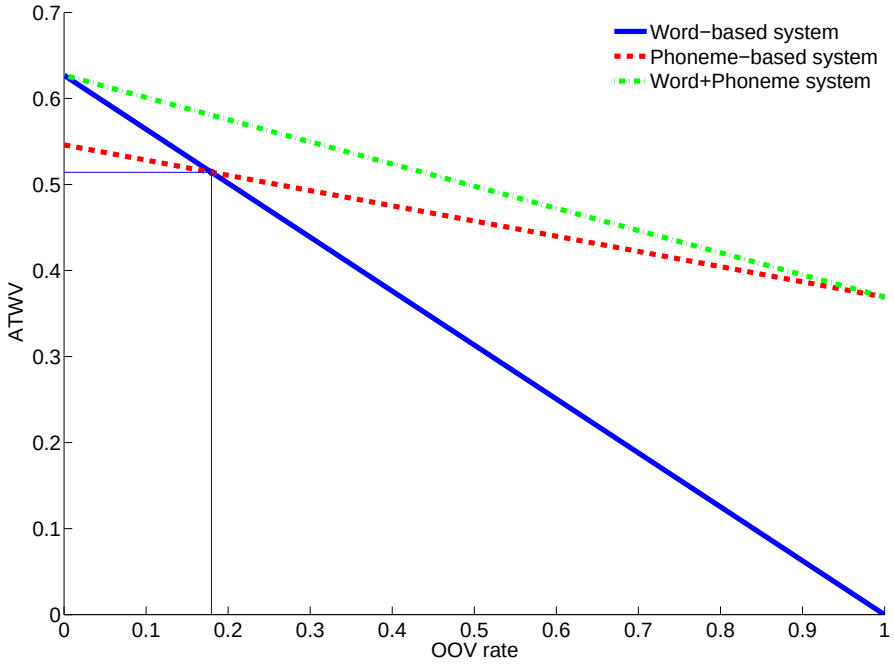

Figure 8.4: The overall performance of the word and phoneme -based systems and their combination, with the OOV rate varying from 0.0 to 1.0.

8.3 Future work

Although the techniques presented in this thesis significantly improve the OOV detection, some problems still remain. For example, unoptimised lattice units and limited understanding of OOV properties. We discuss some future work in this section.

8.3.1 Fast search

Query efficiency is important for a practical STD system. In our study, the term search was performed on lattices and the computation time was linear in the lattice size. This approach is rather inefficient, especially when the lattices are large. To speed up the detection, a fast search should be implemented. A commonly used fast search approach is based on inverted indexes [Allauzen et al., 2004]. Note that the techniques proposed in this thesis can be easily applied to fast search. For example, the SPM-based detection can be conducted on the inverted indexes with no difficulties; the term-dependent discriminative decision can be applied to the searching results; and the direct posterior confidence scores can be stored in the inverted indexes directly.

8.3.2 Term dependent confidence

We have demonstrated that a term-dependent confidence measure improves the decision quality of an STD system. So far we just considered the occurrence-derived factors, leaving many others unexplored, e.g., phonemic structure, morphological structure, tone, syntax role, etc. Especially, distributions of confidence scores of putative detections might convey much information for the hit/FA decision, and hence deserve further study. Recently, Can and Saraçlar [2009] proposed to use the distribution information in detection verification; however their method is not ATWV-oriented.

8.3.3 Variable-sized lattice units

Another interesting study of STD is regarding optimal lattice units. Generally speaking, large units capture more lexical constraints and thus are more powerful in representing INV terms that follow phonological rules, whereas small units are more flexible and thus are suitable for representing OOV terms that are phonologically irregular. This is why we used word lattices to search INV terms and phoneme lattices to search OOV terms. However, INV terms and OOV terms are not absolutely regular or irregular: INV terms are not always rule-followers, and OOV terms are not always exceptions. A type of middle-sized units therefore might be a better choice, e.g., word-fragments [Seide et al., 2004], graphones [Akbacak et al., 2008; Vergyri et al., 2007], multigrams [Pinto et al., 2008; Szöke et al., 2008a], syllables [Mertens and Schneider, 2009], etc.

This idea can be extended to a system based on variable-sized units. In such a

system, a suitable lattice unit would be determined by the regularity of the spelling of the search term, measured according to some linguistic rules or probabilistic models (e.g., joint-multigram model).

8.3.4 System combination

Another interesting topic is system combination, proposed in Section 5.1.4. We have shown that, based on the discriminative confidence measure, detections from heterogeneous systems can be combined according to Equations 6.26-6.31. We also demonstrated that combining the SPM and soft match -based systems gave better performance than each individual system. In fact, there are many different types of systems that can be combined, e.g., systems with various acoustic features, various acoustic units, various order of LMs, etc. Among these combinations, combining phoneme and grapheme -based systems is particularly interesting.

In general, grapheme-based systems exhibit lower performance than phoneme-based systems because less phonetic knowledge is used in acoustic modelling; however, grapheme-based systems possess some inherit advantages. First, a grapheme-based system does not require manually-designed dictionaries, and therefore is easy to implement; second, a grapheme-based system does not need a stochastic model to treat pronunciation variation, since the variation has been modelled within the acoustic models implicitly. An interesting point is that the phoneme and grapheme -based STD systems are highly complementary due to the different ways they model acoustic features. Therefore, we can combine a phoneme-based system with a grapheme-based system to improve STD performance.

The direct posterior confidence approach provides a flexible framework for system combination. Within this framework, term detection and confidence estimation are separated, so we can either combine the phoneme and grapheme-based term detection, or the phoneme and grapheme-based confidence estimation, or both. Our preliminary experiments have demonstrated a promising performance improvement with the phoneme-grapheme combination [Tejedor et al., 2009].

8.3.5 Learning OOV properties

We have proposed a multitude of techniques to deal with the special properties of OOV terms and have achieved encouraging performance improvement; however, our research addresses the OOV properties from just one angle, that is, machine learn-

ing. Research could be conducted from different angles in different disciplines, e.g., linguistics, phonetics or etymology.

For instance, the following questions might be interesting: How are OOV terms different from INV terms in phonetic and/or phonotactic ways? How many possible pronunciations does an OOV term usually have? Does the variation in pronunciation come from phonemic uncertainty or phonetic deviation? Do they cause more complex syntax structure? Do they lead to more ambiguity in semantics? Can we determine the degree of spelling idiosyncrasy? Are these properties domain-dependent? Are their pronunciations adopted to the pronunciation system of the target language, or do they trigger the creation of new pronunciation rules? How many novel words finally make their way into dictionaries and how many fade away? How long does the 'dictionarisation' take, and how does the pronunciation change in this process?

Answering these questions is out of the scope of our current research; nevertheless, findings of linguistic and phonetic studies will certainly help us obtain a deeper insight into OOV properties and construct suitable models to capture them. Representing multi-disciplinary knowledge within a probabilistic model would be an interesting extension to the current work.

8.4 Summary

Looking back at the four questions we raised at the beginning of this thesis, we believe we have found our answers:

1. We can use a joint-multigram model to predict OOV pronunciations;
2. We can use a stochastic pronunciation model to compensate for pronunciation variation of OOV terms;
3. We can use a term-dependent discriminative decision to compensate for OOV diversity;
4. We can use a discriminative model to obtain an OOV-robust confidence measure.

The work in this thesis can not address all the OOV challenges of course; it is just one step towards the direction we believe correct. With more research on OOV issues and other topics of STD, we believe the day when we can freely retrieve information from speech is in the near future.

Bibliography

Dave Abberley, Steve Renals, Gary Cook, and Tony Robinson. Retrieval of broadcast news documents with the thisl system. In *Proc. ICASSP'98*, pages 3781–3784, Seattle, Washington, USA, May 1998.

Gustavo A. Hernández Ábrego. *Confidence Measures for Speech Recognition and Utterance Verification*. PhD thesis, University Politècnica de Catalunya, March 2000.

M.J. Adamson and R.I. Damper. A recurrent network that learns to pronounce English text. In *Proc. ICSLP'96*, pages 1704–1707, Philadelphia, USA, October 1996.

David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

Murat Akbacak and John H.L. Hansen. Spoken proper name retrieval in audio streams for limited-resource language via lattice based search using hybrid representations. In *Proc. ICASSP'06*, volume 1, pages 953–956, Toulouse, France, May 2006a.

Murat Akbacak and John H.L. Hansen. A robust fusion method for multilingual spoken document retrieval systems employing tiered resources. In *Proc. ICSLP'06*, pages 1177–1180, Pittsburgh, USA, September 2006b.

Murat Akbacak, Dimitra Vergyri, and Andreas Stolcke. Open-vocabulary spoken term detection using graphone-based hybrid recognition systems. In *Proc. ICASSP'08*, pages 5240–5243, Las Vegas, Nevada, USA, March 2008.

Cyril Allauzen, Mehryar Mohri, and Murat Saraclar. General indexation of weighted automata application to spoken utterance retrieval. In *Proc. HLT-NAACL 2004*, pages 33–40, Boston, USA, May 2004.

Arnon Amir, Alon Efrat, and Savitha Srinivasan. Advances in phonetic word spotting. In *Proc. The 10th International conference on information and knowledge management (CIKM'01)*, pages 580–582, Atlanta, Georgia, USA, November 2001.

Ove Andersen and Paul Dalsgaard. A self-learning approach to transcription of Danish proper names. In *Proc. ICSLP'94*, pages 1627–1630, Yokohama, Japan, September 1994.

Ove Andersen, Roland Kuhn, Ariane Lazaridès, Paul Dalsgaard, Jürgen Haas, and Elmar Nöth. Comparison of two tree-structured approaches for grapheme-to-phoneme conversion. In *Proc. ICSLP'96*, volume 3, pages 1700–1703, Philadelphia, USA, October 1996.

Hagai Aronowitz. Online vocabulary adaptation using contextual information and information retrieval. In *Proc. Interspeech'08*, pages 1805–1808, Brisbane, Australia, September 2008.

Kartik Audhkhasi and Ashish Verma. Keyword search using modified minimum edit distance measure. In *Proc. ICASSP'07*, volume 4, pages 929–932, Honolulu, Hawaii, USA, April 2007.

Paul C. Bagshaw. Phonemic transcription by analogy in text-to-speech synthesis: Novel word pronunciation and lexicon compression. *Computer Speech & Language*, 12(2):119–142, April 1998.

Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics,*, 37(6):1554–1563, 1966.

Nazife Baykal and Mehmet R. Tolun. An experimental comparison of symbolic and neural learning algorithms. In *Proc. The 2nd International Conference on Knowledge-Based Intelligent Electronic Systems*, volume 2, pages 306–315, Adelaide, SA, Australia, April 1998.

Aruna Bayya. Rejection in speech recognition systems with limited training. In *Proc. ICSLP'98*, Sydney, Australia, November 1998.

Issam Bazzi and James Glass. Heterogeneous lexical units for automatic speech recognition: Preliminary investigations. In *Proc. ICASSP'00*, volume 3, pages 1257–1260, Istanbul, Turkey, June 2000a.

Issam Bazzi and James R. Glass. Modeling out-of-vocabulary words for robust speech recognition. In *Proc. ICSLP'00*, volume 1, pages 401–404, Beijing, China, October 2000b.

Issam Bazzi and James R. Glass. Learning units for domain-independent out-of-vocabulary word modelling. In *Proc. Eurospeech'01*, volume 1, pages 61–64, Aalborg, Denmark, September 2001.

Issam Bazzi and James R. Glass. A multi-class approach for modelling out-of-vocabulary words. In *Proc. ICSLP'02*, pages 1613–1616, Denver, USA, September 2002.

Jerome R. Bellegarda. Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy. *Speech Communication*, 46(12):140–152, June 2005.

Zachary Bergen and Wayne Ward. A senone based confidence measure for speech recognition. In *Proc. Eurospeech 97*, pages 819–822, Rhodes, Greece, September 1997.

Giulia Bernardis and Hervé Bourlard. Improving posterior based confidence measures in hybrid HMM/ANN speech recognition systems. In *Proc. ICSLP'98*, pages 775–778, Sydney, Australia, November 1998.

Klaus Beulen, S. Ortmanns, A. Eiden, S. Martin, L. Welling, J. Overmann, and Hermann Ney. Pronunciation modelling in the RWTH large vocabulary speech recognizer. In *Proc. ESCA Workshop Modeling Pronunciation Variation for Automatic Speech Recognition*, pages 13–16, Kerkrade,Netherlands, May 1998.

J. Billa, M. Noamany, A. Srivastava, D. Liu, R. Stone, J. Xu, J. Makhoul, and F. Kubala. Audio indexing of Arabic broadcast news. In *Proc. ICASSP'02*, pages 5–8, Orlando, Florida, USA, May 2002.

Maximilian Bisani and Hermann Ney. Multigram-based grapheme-to-phoneme conversion for LVCSR. In *Proc. Eurospeech'03*, pages 933–936, Geneva, Switzerland, September 2003a.

Maximilian Bisani and Hermann Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. ICSLP'02*, pages 105–108, Denver, USA, September 2002.

Maximilian Bisani and Hermann Ney. Multigram-based grapheme-to-phoneme conversion for LVCSR. In *Proc. Eurospeech'03*, pages 933–936, Geneva, Switzerland, September 2003b.

Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, May 2008.

Maximilian Bisani and Hermann Ney. Open vocabulary speech recognition with flat hybrid models. In *Proc. Interspeech'05*, pages 725–728, Antwerp, Belgium, August 2005.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Alan W Black, Kevin Lenzo, and Vincernt Pagel. Issues in building general letter to sound rules. In *Proc. 3rd ESCA Workshop on Speech Synthesis*, pages 77–80, Jenolan Caves, Australia, 1998.

Jean-Marc Boite, Herve Bourland, Bart D'hoore, and Marc Haesen. A new approach towards keyword spotting. In *Proc. Eurospeech'93*, pages 1273–1276, Berlin, Germany, September 1993.

Antal Van Den Bosch and Walter Daelemans. Data-oriented methods for grapheme-to-phoneme conversion. In *Proc. 6th conference on European chapter of the Association for Computational Linguistics (EACL)*, pages 45–53, Utrecht, The Netherlands, 1993.

Louis Ten Bosch, Annika Hämäläinen, Odette Scharenborg, and Lou Boves. Acoustic scores and symbolic mismatch penalties in phone lattices. In *Proc. ICASSP'06*, pages 437–440, Toulouse, France, May 2006.

Hervé Bourlard, Bart D'hoore, and Jean-Marc Boite. Optimizing recognition and rejection performance in wordspotting systems. In *Proc. ICASSP'94*, volume 1, pages 373–376, Adelaide, SA, Australia, April 1994.

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman & Hall;, January 1984.

J. S. Bridle. An efficient elastic-template method for detecting given words in running speech. In *Proc. of the British Acoustic Society Meeting*, April 1973.

Martin Brown, Jonathan Trumbull Foote, Gareth J. F. Jones, Karen Spärck Jones, and Steve Young. Open-vocabulary speech indexing for voice and video mail retrieval. In *Proc. ACM Multimedia conference*, Boston, MA, 1996.

Susanne Burger, Victoria MacLaren, and Hua Yu. The ISL meeting corpus: the impact of meeting type on speech style. In *Proc. ICSLP'02*, pages 301–304, Denver, USA, September 2002.

Lukáš Burget, Jan Černocký, Michal Fapšo, Martin Karafiát, Pavel Matějka, Petr Schwarz, Pavel Smrž, and Igor Szöke. Indexing and search methods for spoken documents. In *Text, Speech and Dialogue*, volume 4188/2006 of *Lecture Notes in Computer Science*, pages 351–358. Springer Berlin / Heidelberg, 2006.

Dogan Can, Erica Cooper, Abhinav Sethy, Chris White, Bhuvana Ramabhadran, and Murat Saraclar. Effect of pronunciations on OOV queries in spoken term detection. In *Proc. ICASSP'09*, pages 3957–3960, Taipei, Taiwan, April 2009.

Doğan Can and Murat Saraçlar. Score distribution based term specific thresholding for spoken term detection. In *Proc. NAACL HLT 2009*, pages 269–272, Boulder, Colorado, June 2009.

Peter S. Cardillo, Mark Clements, and Michael S. Miller. Phonetic searching vs. LVCSR: How to find what you really want in audio archives. *International Journal of Speech Technology*, 5(1):9–22, January 2002.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Lin Chase. Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. Eurospeech'97*, pages 815–818, Rhodes, Greece, September 1997.

Upendra Chaudhari, Hong-Kwang Jeff Kuo, and Brian Kingsbury. Discriminative graph training for ultra-fast low-footprint speech indexing. In *Proc. Interspeech 2008*, pages 2175–2178, Las Vegas, Nevada, USA, March 2008.

Ciprian Chelba and Alex Acero. Position specific posterior lattices for indexing speech. In *Proc. ACL 2005*, pages 443–450, Ann Arbor, Michigan, June 2005.

Stanley F. Chen. Conditional and joint models for grapheme-to-phoneme conversion. In *Proc. Eurospeech'03*, pages 2033–2036, Geneva, Switzerland, September 2003.

Noam Chomsky and Morris Halle. *The Sound Pattern of English*. Harper & Row, New York, 1968.

Richard W. Christiansen and Craig K. Rushforth. Detecting and locating key words in continuous speech using linear predictive coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(5):361–367, October 1977.

Rob Clark, Korin Richmond, and Simon King. MULTISYN: Open-domain unit selection for the Festival speech synthesis system. *Speech Communication*, 49(4): 317–330, April 2007.

Stephen Cox and Richard Rose. Confidence measures for the SWITCHBOARD database. In *Proc. ICASSP'96*, volume 1, pages 511–514, Atlanta, Georgia, USA, May 1996.

Nick Cremelie and Jean-Pierre Martens. In search of better pronunciation models for speech recognition. *Speech Communication*, 29(2-4):115–136, 1999.

Catia Cucchiarini, Helmer Strik, and Lou Boves. Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithms. *Speech Communication*, 30(2-3):109–119, 2000.

Walter Daelemans and Antal van den Bosch. TabTalk: Reusability in data-oriented grapheme-to-phoneme conversion. In *Proc. Eurospeech'93*, pages 1459–1462, Berlin, Germany, September 1993.

Walter Daelemans and Antal van den Bosch. Language-independent data-oriented grapheme-to-phoneme conversion. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in speech synthesis*, pages 77–89. Springer-Verlag, 1996.

Walter Daelemans, Antal van den Bosch, and Ton Weijters. IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review, Special Issue on Lazy Learning*, 11:407–423, 1997.

Walter Daelemans, Antal Van den Bosch, Jakub Zavrel, Jorn Veenstra, Sabine Buchholz, and Bertjan Busser. Rapid development of NLP modules with memory-based learning. In *Proc. ELSNET 98*, pages 105–113, Wonderland, March 1998.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–41, 1999.

R.I. Damper and J.F.G. Eastmond. Pronunciation by analogy: Impact of implementational choices on performance. *Language and Speech*, 40(1):1–23, 1997.

R.I. Damper, Y. Marchand, M.J. Adamson, and K. Gustafson. A comparison of letter-to-sound conversion techniques for English text-to-speech synthesis. *Proc. of the Institute of Acoustics*, 20(6):245–254, 1998.

Bart Decadt, Jacques Duchateau, Walter Daelemans, and Patrick Wambacq. Phoneme-to-grapheme conversion for out-of-vocabulary words in large vocabulary speech recognition. In *Proc. ASRU'01*, pages 413–416, Madonna di Campiglio Trento Italy, December 2001.

Michael J. Dedina and Howard C. Nusbaum. PRONOUNCE: A program for pronunciation by analogy. Technical Report 12, Indiana University, Bloomington, Indiana, 1986. Speech Research Laboratory Progress Report 12.

Sabine Deligne, Francois Yvon, and Frédéric Bimbot. Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc. Eurospeech'95*, pages 2243–2246, Madrid, Spain, September 1995.

Satya Dharanipragada and Salim Roukos. Experimental results in audio-indexing. In *Proc. DARPA 1997 Speech Recognition Workshop*, Chantilly, VA, 1997.

Satya Dharanipragada and Salim Roukos. A multistage algorithm for spotting new words in speech. *IEEE Transactions on Speech and Audio Processing*, 10(8):542–550, November 2002.

Thomas G. Dietterich, Hermann Hild, and Ghulum Bakiri. A comparison of ID3 and backpropagation for English text-to-speech mapping. *Machine Learning*, 18(1):51–80, January 1995.

Michel Divay and Anthony J. Vitale. Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis. *Computational Linguistics*, 23(4):495–523, 1997.

Corentin Dubois and Delphine Charlet. Using textual information from LVCSR transcripts for phonetic-based spoken term detection. In *Proc. ICASSP'08*, pages 4961–4964, Las Vegas, Nevada, USA, March 2008.

Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *Proc. International Conference on Artificial Neural Networks (ICANN 2007)*, volume 4669/2007 of *Lecture Notes in Computer Science*, pages 220–229. Springer Verlag, 2007.

Pablo Fetter, Frédéric Dandurand, and Peter Regel-Brietzmann. Word graph rescoring using confidence measures. In *Proc. ICSLP'96*, pages 10–13, Philadelphia, USA, October 1996.

Jonathan Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. ASRU'97*, 1997.

Jonathan Fiscus, Jérôme Ajot, and George Doddington. English STD 2006 results. Technical report, National Institute of Standards and Technology (NIST), 2006. URL http://www.nist.gov/speech/publications/presentations/.

Jonathan G. Fiscus, Jerome Ajot, John S. Garofolo, and George Doddingtion. Results of the 2006 spoken term detection evaluation. In *Proc. Workshop on Searching Spontaneous Conversational Speech (SIGIR-SSCS'07)*, Amsterdam, July 2007.

Susan Fitt. Documentation and user guide to UNISYN lexicon and post-lexical rules. Technical report, Centre for Speech Technology Research, Edinburgh, 2000.

Jonathan Trumbull Foote, Gareth J. F. Jones, Karen Sparck Jones, and Steve J. Young. Talker-independent keyword spotting for information retrieval. In *Proc. Eurospeech'95*, pages 2145–2148, Madrid, Spain, September 1995.

G. David Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

Joe Frankel, Dong Wang, and Simon King. Growing bottleneck features for tandem ASR. In *Proc. Interspeech'08*, page 1549, September 2008.

Toshiaki Fukada and Yoshinori Sagisaka. Automatic generation of a pronunciation dictionary based on a pronunciation network. In *Proc. Eurospeech'97*, pages 2471–2474, Rhodes, Greece, September 1997.

Lucian Galescu. Recognition of out-of-vocabulary words with sub-lexical language models. In *Proc. Eurospeech'03*, pages 249–252, Geneva, Switzerland, September 2003.

Lucian Galescu and James F. Allen. Pronunciation of proper names with a joint n-gram model for bi-directional grapheme-to-phoneme conversion. In *Proc. ICSLP'02*, volume 1, pages 109–112, Denver, USA, September 2002.

Lucian Galescu and James F. Allen. Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *Proc. 4th ISCA ITRW on Speech Synthesis (SSW4-2001)*, Perthshire, Scotland, August 2001.

Larry Gillick, Yoshiko Ito, and Jonathan Young. A probabilistic approach to confidence estimation and evaluation. In *Proc. ICASSP'97*, pages 879–882, Munich, Bavaria, Germany, April 1997.

Herbert Gish, Kenney Ng, and J. Robin Rohlicek. Secondary processing using speech segments for an HMM word spotting system. In *Proc. ICSLP'92*, pages 17–20, Banff, Canada, October 1992.

Robert J. Glushko. The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance*, 5(4):674–691, 1979.

David Grangier. *Machine learning for information retrieval*. PhD thesis, University of Nice Sophia Antipolis, France, 2008.

Thomas Hain. *Hidden Model Sequence Models for Automatic Speech Recognition*. PhD thesis, University of Cambridge, November 2001.

Thomas Hain. Implicit pronunciation modelling in ASR. In *Proc. ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexical Adaptation for Spoken Language*, Estes Park, CO, 2002.

Thomas Hain and Phil Woodland. Dynamic HMM selection for continuous speech recognition. In *Proc. Eurospeech'99*, volume 3, pages 1327–1330, Budapest, Hungary, September 1999.

Thomas Hain, Lukas Burget, John Dines, Iain McCowan, Martin Karafiat, Mike Lincoln, Darren Moore, Giulia Garau, Vincent Wan, Roeland Ordelman, and Steve Renals. The development of the AMI system for the transcription of speech in meetings. In *Proc. MLMI'05*, Edinburgh, 2005.

Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Iain McCowan, Darren Moore, Vincent Wan, Roeland Ordelman, and Steve Renals. The 2005 AMI system for the transcription of speech in meetings. In *Machine Learning for Multimodal Interaction*, volume 3869/2006, pages 450–462. Springer Berlin /Heidelberg, 2006a.

Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Jithendra Vepa, and Vincent Wan. The AMI meeting transcription system: Progress and performance. In *Machine Learning for Multimodal Interaction*, volume 4299/2006, pages 419–431. Springer Berlin/Heidelberg, 2006b.

Juha Häkkinen, Janne Suontausta, Soren Riis, and Kare Jean Jensen. Assessing text-to-phoneme mapping strategies in speaker independent isolated word recognition. *Speech Communication*, 41(2):455–467, 2003.

William I. Hallahan. DECTalk Software: Text-to-speech technology and implementation. *Digital Technical Journal*, 7(4):5–19, 1995.

A. G. Hauptmann, R. E. Jones, K. Seymore, S. T. Slattery, M. J. Witbrock, and M. A. Siegler. Experiments in information retrieval from spoken documents. In *Proc. DARPA Workshop on Broadcast News Transcription and Understanding*, pages 175–181, Lansdowne VA, February 1998.

Hynek Hermansky, Daniel P.W. Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP'00*, pages 1635–1638, Istanbul, Turkey, June 2000.

Alan L. Higgins and Robert E. Wohlford. Keyword recognition using template concatenation. In *Proc. ICASSP'85*, pages 1233–1236, Tampa, Florida, USA, March 1985.

J. Hochberg, S.M. Mniszewski, T. Calleja, and G.J. Papcun. A default hierarchy for pronouncing English. *IEEE translations on Pattern Analysis and Machine Intelligence*, 13(9):957–964, September 1991.

Qian Hu, Fred Goodman, Stanley Boykin, Randy Fish, and Warren Greiff. Audio hot spotting and retrieval using multiple features. In *HLT-NAACL 2004 Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, pages 13–17, Boston Common, May 2004.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001.

Man hung Siu, Herbert Gish, and Fred Richardson. Improved estimation, evaluation and applications of confidence measures for speech recognition. In *Proc. Eurospeech'97*, pages 831–834, Rhodes, Greece, September 1997.

Yoshiaki Itoh, Takayuki Otake, Kohei Iwata, Kazunori Kojima, Masaaki Ishigame, Kazuyo Tanaka, and Shi wook Lee. Two-stage vocabulary-free spoken document retrieval-subword identification and re-recognition of the identified sections. In *Proc. ICSLP'06*, pages 1161–1164, Pittsburgh, USA, September 2006.

Kenji Iwata, Koichi Shinoda, and Sadaoki Furui. Robust spoken term detection using combination of phone-based and word-based recognition. In *Proc. Interspeech'08*, pages 2195–2198, Brisbane, Australia, September 2008.

David A. James. A system for unrestricted topic retrieval from radio news broadcasts. In *Proc. ICASSP'96*, pages 279–282, Atlanta, Georgia, USA, May 1994.

David A. James. A system for unrestricted topic retrieval from radio news broadcasts. In *Proc. ICASSP'96*, volume 1, pages 279–282, Atlanta, Georgia, USA, May 1996.

David A. James and Steve J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proc. ICASSP'94*, pages 377–380, Yokohama, Japan, September 1994.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. The ICSI meeting corpus. In *Proc. ICASSP'03*, pages 364–367, Hong Kong, April 2003.

A. Jansen and P. Niyogi. Point process models for spotting keywords in continuous speech. Technical report, University of Chicago, September 2008.

P. Jeanrenaud, K. Ng, M. Siu, J.R. Rohlicek, and H. Gish. Phonetic-based word spotter: various configurations and application to event spotting. In *Proc. Eurospeech'93*, pages 1057–1060, Berlin, Germany, September 1993.

P. Jeanrenaud, M. Siu, and Herbert Gish. Large vocabulary word scoring as a basis for transcription generation. In *Proc. Eurospeech'95*, pages 2149–2152, Madrid, Spain, September 1995.

Kare Jean Jensen and Soren Riis. Self-organizing letter code-book for text-to-phoneme neural network model. In *Proc. ICSLP'00*, volume 3, pages 318–321, Beijing, China, October 2000.

Hui Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4):455–470, December 2005.

Li Jiang, Hsiao-Wuen Hon, and Xuedong Huang. Improvements on a trainable letter-to-sound converter. In *Proc. Eurospeech'97*, volume 2, pages 605–608, Rhodes, Greece, September 1997.

Gareth J F Jones, Jonathan Trumbull Foote, Karen Sparck Jones, and Steve J. Young. Robust talker-independent audio document retrieval. In *Proc. ICASSP'96*, pages 311–314, Atlanta, Georgia, USA, May 1996a.

Gareth J. F. Jones, Jonathan Trumbull Foote, Karen Spärck Jones, and Steve J. Young. Retrieving spoken documents by combining multiple index sources. In *Proc. ACM SIGIR'96*, pages 30–38, Zurich Switzerland, August 1996b.

J. Junkawitsch, L. Neubauer, H. Höge, and G. Ruske. A new keyword spotting algorithm with pre-calculated optimal thresholds. In *Proc. ICSLP'06*, pages 2067–2070, Pittsburgh, USA, September 1996.

Simo O. Kamppari and Timothy J. Hazen. Word and phone level acoustic confidence scoring. In *Proc. ICASSP'00*, volume 3, pages 1799–1802, Istanbul, Turkey, June 2000.

Ronald M. Kaplan and Martin Kay. Regular model of phonological rule systems. *Computational Linguistics*, 20(3):331–378, September 1994.

Thomas Kemp and Thomas Schaaf. Estimating confidence using word lattices. In *Proc. Eurospeech'97*, pages 827–830, Rhodes, Greece, September 1997.

Thomas Kemp and Alex Waibel. Reducing the OOV rate in broadcast news speech recognition. In *Proc. ICSLP'98*, pages 1839–1842, Sydney, Australia, November 1998.

Joseph Keshet, David Grangier, and Samy Bengio. Discriminative keyword spotting. *Speech Communication*, 51(4):317–329, April 2009.

Judith M. Kessens, Mirjam Wester, and Helmer Strik. Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation. *Speech Communication*, 29(2-4):193–207, 1999.

Hamed Ketabdar, Jithendra Vepa, Samy Bengio, and Hervé Bourlard. Posterior based keyword spotting with a priori thresholds. In *Proc. ICSLP'06*, pages 1642–1645, Pittsburgh, USA, September 2006.

Anne K. Kienappel and Reinhard Kneser. Designing very compact decision trees for grapheme-to-phoneme transcription. In *Proc. Eurospeech'01*, pages 1911–1914, Aalborg, Denmark, September 2001.

Byeongchang Kim, Geunbae Lee, and Jong-Hyeok Lee. Hybrid grapheme to phoneme conversion for unlimited vocabulary. *Natural Language Engineer*, 1998.

Dietrich Klakow, Georg Rose, and Xavier Aubert. OOV-detection in large vocabulary system using automatically defined word-fragments as fillers. In *Proc. Eurospeech'99*, pages 49–52, Budapest, Hungary, September 1999.

Dennis H. Klatt. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82(3):737–793, 1987.

Dennis H. Klatt and David W. Shipman. Letter-to-phoneme rules: A semi-automatic discovery procedure. *Journal of the Acoustical Society of America*, 72(s1):s48, November 1982.

Hui Lin, Alex Stupakov, and Jeff Bilmes. Spoken keyword spotting via multi-lattice alignment. In *Proc. Interspeech'08*, pages 2191–2194, Brisbane, Australia, September 2008.

Hui Lin, Alex Stupakov, and Jeff Bilmes. Improving multi-lattice alignment based spoken keyword spotting. In *Proc. ICASSP'09*, pages 4877–4880, Taipei, Taiwan, April 2009.

Eduardo Lleida, Jose B. Mari no, Josep Salavedra, Antonio Bonafonte, E. Monte, and A. Martínez. Out-of-vocabulary word modelling and rejection for keyword spotting. In *Proc. Eurospeech'93*, pages 1265–1268, Berlin, Germany, September 1993.

Beth Logan and JM Van Thong. Confusion-based query expansion for OOV words in spoken document retrieval. In *Proc. ICSLP'02*, pages 1997–2000, May, September 2002.

Beth Logan, Pedro Moreno, Jean-Manuel Van Thong, and Ed Whittaker. An experimental study of an audio indexing system for the web. In *Proc. ICSLP'00*, volume 2, pages 676–679, Beijing, China, October 2000.

Beth Logan, Pedro Moreno, and Om Deshmuk. Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio. In *Proc. HLT'02*, pages 31–35, San Francisco, 2002.

Beth Logan, Jean-Manuel Van Thong, and Pedro J. Moreno. Approaches to reduce the effects of OOV queries on indexed spoken audio. *IEEE Transaction on Multimedia*, 7(5):899–906, October 2005.

J.M. Lucassen and R.L. Mercer. An information theoretic approach to the automatic determination of phonetic baseforms. In *Proc. ICASSP'84*, pages 42.5.1–42.5.4, San Diego, California, USA, March 1984.

R.W.P. Luk and R.I. Damper. Stochastic phonographic transduction for English. *Computer Speech and Language*, 10:133–153, 1996.

Bin Ma and Haizhou Li. A phonotactic-semantic paradigm for automatic spoken document classification. In *Proc. 28th international ACM SIGIR conference on Research and development in information retrieval*, pages 369–376, Salvador, Brazil, August 2005.

Changxue Ma and Mark A. Randolph. An approach to automatic phonetic baseform generation based on Bayesian networks. In *Proc. Eurospeech'01*, pages 1453–1457, Aalborg, Denmark, September 2001.

Chengyuan Ma and Chin-Hui Lee. A study on word detector design and knowledge-based pruning and rescoring. In *Proc. Interspeech'07*, pages 1473 –1476, Antwerp, Belgium, August 2007.

Jonathan Mamou and Bhuvana Ramabhadran. Phonetic query expansion for spoken document retrieval. In *Proc. Interspeech'08*, pages 2106–2109, Brisbane, Australia, September 2008.

Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. Vocabulary independent spoken term detection. In *Proc. ACM-SIGIR'07*, pages 615–622, 2007.

Jonathan Mamou, Yosi Mass, Bhuvana Ramabhadran, and Benjamin Sznajder. Combination of multiple speech transcription methods for vocabulary independent search. In *Proc. Workshop on Search in Spontaneous Conversational Speech (SIGIR-SSCS'08)*, Singapore, 2008.

Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Proc. Eurospeech'99*, pages 495–498, Budapest, Hungary, September 1999.

Alexandros Manos and Victor Zue. A segment-based wordspotter using phonetic filler models. In *Proc. ICASSP'97*, volume 2, pages 899–902, Munich, Bavaria, Germany, April 1997.

Jean manuel Van Thong, David Goddeau, Anna Litvinova, Beth Logan, Pedro Moreno, and Michael Swain. Speechbot: A speech recognition based audio indexing system for the web. In *Proc. The 6th RIAO Conference, 2000*, pages 106–115, Paris, April 2000.

Yannick Marchand and Robert I. Damper. A multistrategy approach to improving pronunciation by analogy. *Computational Linguistics*, 26(2):195–219, June 2000.

Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The DET curve in assessment of detection task performance. In *Proc. Eurospeech'97*, volume 4, pages 1895–1898,, Rhodes, Greece, September 1997.

Luc Mathan and Laurent Miclet. Rejection of extraneous input in speech recognition applications using multi-layer perceptrons and the trace of HMMs. In *Proc. ICASSP'91*, volume 1, pages 93–96, Toronto, Ont., Canada, May 1991.

Neil McCulloch, Mark Bedworth, and John Bridle. NETspeak – A re-implementation of NETtalk. *Computer Speech and Language*, 2:289–301, 1987.

J. McDonough, K. Ng, P. Jeanrenaud, H. Gish, and J.R. Rohlicek. Approaches to topic identification on the SWITCHBOARD corpus. In *Proc. ICASSP'94*, volume 1, pages 385–388, Adelaide, SA, Australia, April 1994.

Rachida El Méliani and Douglas O'Shaughnessy. Accurate keyword spotting using strictly lexical fillers. In *Proc. ICASSP'97*, pages 907–910, Munich, Bavaria, Germany, April 1997.

Helen Meng. A hierarchical lexical representation for bi-directional spelling-to-pronunciation/pronunciation-to-spelling generation. *Speech Communication*, 33(3): 213–239, February 2001.

Helen Meng, Stephanie Seneff, and Victor Zue. Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation. In *Proc. Workshop on Human Language Technology (HLT'94)*, pages 289–294, Plainsboro, NJ, March 1994.

Helen Meng, Sheri Hunnicut, Stephanie Seneff, and Victor Zue. Reversible letter-to-sound/sound-to-letter generation based on parsing word morpology. *Speech Communication*, 18(1):47–63, 1996.

Sha Meng, Peng Yu, Frank Seide, and Jia Liu. A study of lattice-based spoken term detection for Chinese spontaneous speech. In *Proc. ASRU'07*, pages 635–640, Kyoto, Japan, December 2007.

Sha Meng, Jian Shao, Roger Peng Yu, Jia Liu, and Frank Seide. Addressing the out-of-vocabulary problem for large-scale Chinese spoken term detection. In *Proc. Interspeech'08*, pages 2146–2149, Brisbane, Australia, September 2008a.

Sha Meng, Peng Yu, Jia Liu, , and Frank Seide. Fusing multiple systems into a compact lattice index for Chinese spoken term detection. In *Proc. ICASSP'08*, pages 4345–4348, Las Vegas, Nevada, USA, March 2008b.

Timo Mertens and Daniel Schneider. Efficient subword lattice retrieval for German spoken term detection. In *Proc. ICASSP'09*, pages 4885–4888, 2009.

David R. H. Miller, Michael Kleber, Chia lin Kao, Owen Kimball, Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, and Herbert Gish. Rapid and accurate spoken term detection. In *Proc. Interspeech'07*, pages 314–317, Antwerp, Belgium, August 2007.

Mohamed, Renato De Mori, Olivier Deroo, Staphane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, Richard Rose, Vivek Tyagi, and Christian Wellekens. Automatic speech recognition and intrinsic speech variation. In *Proc. ICASSP'06*, volume 5, pages 1021–1024, Toulouse, France, May 2006.

C. S. Myers, Lawrence R. Rabiner, and A. E. Rosenberg. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. In *Proc. ICASSP'80*, pages 173–177, Denver, Colorado, USA, April 1980.

Chalapathy V. Neti, Salim Roukos, and E. Eide. Word-based confidence measures as a guide for stack search in speech recognition. In *Proc. ICASSP'97*, pages 883–886, Munich, Bavaria, Germany, April 1997.

Kenney Ng. *Subword-based Approaches for Spoken Document Retrieval*. PhD thesis, MIT, February 2000.

Kenney Ng. Towards robust methods for spoken document retrieval. In *Proc. ICSLP'98*, pages 939–942, Sydney, Australia, November 1998.

NIST. *Spring 2007 (RT-07) Rich Transcription Meeting Recognition Evaluation Plan*. National Institute of Standards and Technology, Gaithersburg, MD, USA, 2 edition, Feburary 2007. URL http://www.nist.gov/speech/tests/rt/2007/docs/rt07-meeting-eval-plan-v2.pdf.

NIST. *The spoken term detection (STD) 2006 evaluation plan*. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 10 edition, September 2006. URL http://www.nist.gov/speech/tests/std.

Yoo Rhee Oh, Jae Sam Yoon, and Hong Kook Kim. Acoustic model adaptation based on pronunciation variability analysis for non-native speech recognition. *Speech Communication*, 49(1):59–70, 2007.

Vincent Pagel, Kevin Lenzo, and Alan W. Black. Letter-to-sound rules for accented lexicon compression. In *Proc. ICSLP'98*, volume 5, pages 2015–2018, Sydney, Australia, November 1998.

S. H. Parfitt and R. A. Sharman. A bi-directional model of English pronunciation. In *Proc. Eurospeech'91*, pages 801–804, Genoa, Italy, September 1991.

Siddika Parlak and Murat Saraçlar. Spoken term detection for Turkish broadcast news. In *Proc. ICASSP'08*, pages 5244–5247, Las Vegas, Nevada, USA, March 2008.

Joel Pinto, Igor Szöke, S.R.M. Prasanna, and Hynek Heřmanský. Fast approximate spoken term detection from sequence of phonemes. In *Proc. The 31st Annual International ACM SIGIR Conference*, pages 28–33, Singapore, July 2008. Association for Computing Machinery. ISBN 978-90-365-2697-5.

John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, February 1992.

Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

Mazin G Rahim, Chin-Hui Lee, and Biing-Hwang Juang. Robust utterance verification for connected digits recognition. In *Proc. ICASSP'95*, volume 1, pages 285–288, Detroit, Michigan, USA, May 1995.

Mazin G. Rahim, Chin-Hui Lee, and Biing-Hwang Juang. Discriminative utterance verification for connected digits recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3):266–277, May 1997.

Bhuvana Ramabhadran, Lalit R. Bahl, Peter V DeSouza, and Mukund Padmanabhan. Acoustics-only based automatic phonetic baseform generation. In *Proc. ICASSP'98*, volume 1, pages 309–312, Seattle, Washington, USA, May 1998.

Bhuvana Ramabhadran, Abhinav Sethy, Jonathan Mamou, Brian Kingsbury, and Upendra Chaudhari. Fast decoding for open vocabulary spoken term detection. In *Proc. NAACL HLT 2009*, pages 277–280, Boulder, Colorado, June 2009.

Michael Riley, William Byrne, Michael Finke, Sanjeev Khudanpur, Andrej Ljolje, John McDonough, Harriet Nock, Murat Saraclar, Charles Wooters, and George Zavaliagkos. Stochastic pronunciation modelling from hand-labelled phonetic corpora. *Speech Communication*, 29(2-4):209–224, 1999.

Ze'ev Rivlin. A confidence measure for acoustic likelihood scores. In *Proc. Eurospeech'95*, pages 523–526, Detroit, Michigan, USA, May 1995.

Ze'ev Rivlin, Michael Cohen, Victor Abrash, and Thomas Chung. A phone-dependent confidence measure for utterance rejection. In *Proc. ICASSP'96*, volume 1, pages 515–517, Atlanta, Georgia, USA, May 1996.

J. Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish. Continuous hidden Markov modeling for speaker-independent word spotting. In *Proc. ICASSP'89*, volume 1, pages 627–630, Glasgow, UK, May 1989a.

J. Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish. Continuous hidden Markov modeling for speaker-independent word spotting. In *Proc. ICASSP'89*, pages 627–630, Glasgow, UK, May 1989b.

Richard C. Rose. Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech. In *Proc. ICASSP92*, volume 2, pages 105–108, San Francisco, CA, USA, March 1992.

Richard C. Rose and Douglas B. Paul. A hidden Markov model based keyword recognition system. In *Proc. ICASSP'90*, pages 129–132, Albuquerque, NM, USA, April 1990.

Richard C. Rose, Biing-Hwang Juang, and Chin-Hui Lee. A training procedure for verifying string hypotheses in continuous speech recognition. In *Proc. ICASSP'95*, pages 281–284, Detroit, Michigan, USA, May 1995.

Bernhard Rueber. Obtaining confidence measures from sentence probabilities. In *Proc. Eurospeech'97*, pages 739–742, Rhodes, Greece, September 1997.

Murat Saraclar and Richard Sproat. Lattice-based search for spoken utterance retrieval. In *Proc. HLT-NAACL 2004*, pages 129–136, Boston, USA, May 2004.

Murat Saraclar, Harriet Nock, and Sanjeev Khudanpur. Pronunciation modeling by sharing Gaussian densities across phonetic models. In *Proc. Eurospeech'99*, Budapest, Hungary, September 1999.

Thomas Schaaf and Thomas Kemp. Confidence measures for spontaneous speech recognition. In *Proc. ICASSP'97*, pages 875–878, Munich, Bavaria, Germany, April 1997.

Peter Schäuble and Martin Wechsler. First experiences with a system for content based retrieval of information from speech recordings. In *Proc. Workshop on Intelligent Multimedia Information Retrieval (IJCAI'95)*, pages 59–69, Montreal, Quebec, Canada, August 1995.

Frank Seide, Peng Yu, Chengyuan Ma, , and Eric Chang. Vocabulary-independent search in spontaneous speech. In *Proc. ICASSP'04*, volume 1, pages 253–256, Montreal, Quebec, Canada, May 2004.

Frank Seide, Kit Thambiratnam, and Roger Peng Yu. Word-lattice based spoken-document indexing with standard text indexers. In *Proc. SLT 2008*, pages 293–296, Goa, India, December 2008.

Terrence J. Sejnowski and Charles R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1(1):145–168, 1987.

Anand R. Setlur, Rafid A. Sukkar, and John Jacob. Correcting recognition errors via discriminative utterance verification. In *Proc.ICSLP'96*, pages 602–605, Philadelphia, USA, October 1996.

Zak Shafran, Brian Roark, and Seeger Fisher. OGI spoken term detection system. In *Proc. NIST spoken term detection workshop (STD 2006)*, Gaithersburg, Maryland, USA, December 2006.

Jude W. Shavlik, Raymond J. Mooney, and Geoffrey G. Towell. Symbolic and neural learning algorithms: an experimental comparison. *Machine Learning*, 6(2):111–143, March 1991.

Wade Shen, Christopher M. White, and Timothy J. Hazen. A comparison of query-by-example methods for spoken term detection. In *Proc. Interspeech'09*, pages 2143–2146, Brighton, UK, September 2009.

Marius-Călin Silaghi and Hervé Bourlard. Iterative posterior-based keyword spotting without filler models. In *Proc. ASRU'99*, Keystone, Colorado, December 1999.

Olivier Siohan and Michiel Bacchiani. Fast vocabulary-independent audio search using path-based graph indexing. In *Proc. Eurospeech'05*, pages 53–56, Lisbon, Portugal, September 2005.

Manhung Siu and Herbert Gish. Evaluation of word confidence for speech recognition systems. *Computer Speech and Language*, 13(4):299–319, 1999.

Tilo Sloboda and Alex Waibel. Dictionary learning for spontaneous speech recognition. In *Proc. ICSLP'96*, pages 2328–2331, Philadelphia, USA, October 1996.

Richard Sproat, Jim Baker, Martin Jansche, Bhuvana Ramabhadran, Michael Riley, Murat Saraçlar, Abhinav Sethy, Patrick Wolfe, Sanjeev Khudanpur, Arnab Ghoshal, Kristy Hollingshead, Chris White, Ting Qian, Erica Cooper, and Morgan Ulinski. Multilingual spoken term detection: Finding and testing new pronunciations. Technical report, JHU, 2008.

Savitha Srinivasan and Dragutin Petkovic. Phonetic confusion matrix based spoken document retrieval. In *Proc. The 23rd annual international ACM SIGIR conference*

on Research and development in information retrieval (SIGIR'00), pages 81–87, New York, NY, USA, 2000.

Craig Stanfill and David Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, December 1986.

Andreas Stolcke. SRILM – An extensible language modeling toolkit. In *Proc. ICSLP'02*, pages 901–904, Denver, USA, September 2002.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. Discriminative named entity recognition of speech data using speech recognition confidence. In *Proc. ICSLP'06*, pages 1153–1156, Pittsburgh, USA, September 2006.

Rafid A. Sukkar. Subword-based minimum verification error (SB-MVE) training for task independent utterance verification. In *Proc. ICASSP'98*, pages 229–232, Seattle, Washington, USA, May 1998.

Rafid A. Sukkar and Chin-Hui Lee. Vocabulary independent discriminative utterance verfication for nonkeyword rejection in subword based speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(6):420–429, November 1996.

Rafid A. Sukkar and Jay G. Wilpon. A two pass classifier for utterance rejection in keyword spotting. In *Proc. ICASSP'93*, volume 2, pages 451–454, Minneapolis, MN, USA, April 1993.

Rafid A. Sukkar, Anand R. Setlur, Mazin G. Rahim, and Chin-Hui Lee. Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training. In *Proc. ICASSP'96*, pages 518–521, Atlanta, Georgia, USA, May 1996.

K.P.H. Sullivan and R.I. Damper. Novel-word pronunciation within a text-to-speech system. In *Proc. ESCA Workshop on Speech Synthesis*, pages 97–100, Autrans, France, September 1990.

Janne Suontausta and Juha Häkkinen. Decision tree based text-to-phoneme mapping for speech recognition. In *Proc. ICSLP'00*, pages 831–834, Beijing, China, October 2000.

Igor Szöke, Petr Schwarz, Pavel Matějka, Lukáš Burget, Martin Karafiát, Michal Fapšo, and Jan Černocký. Comparison of keyword spotting approaches for informal continuous speech. In *Proc. Interspeech'05*, pages 633–636, Lisbon, Portugal, September 2005a.

Igor Szöke, Petr Schwarz, Pavel Matějka, Lukáš Burget, Martin Karafiát, and Jan Černocký. Phoneme based acoustics keyword spotting in informal continuous speech. In *Proc. TSD 2005*, volume 3658/2005 of *Lecture Notes in Computer Science*, pages 302–309. Springer Berlin / Heidelberg, 2005b.

Igor Szöke, Michal Fapšo, Martin Karafiát, Lukáš Burget, František Grézl, Petr Schwarz, Ondřej Glembek, Pavel Matějka, Stanislav Kontár, and Jan Černocký. BUT system for NIST STD 2006 - English. In *Proc. NIST Spoken Term Detection Evaluation workshop (STD'06)*, Gaithersburg, Maryland, USA, December 2006. National Institute of Standards and Technology.

Igor Szöke, Lukáš Burget, Jan Černocký, and Michal Fapšo. Sub-word modeling of out of vocabulary words in spoken term detection. In *Proc. IEEE Workshop on Spoken Language Technology (SLT'08)*, pages 273–276, Goa, India, December 2008a. ISBN 978-1-4244-3472-5.

Igor Szöke, Michal Fapšo, Lukáš Burget, and Jan Černocký. Hybrid word-subword decoding for spoken term detection. In *Proc. Speech search workshop at SIGIR (SSCS'08)*, Singapore, 2008b. Association for Computing Machinery. ISBN 978-90-365-2697-5.

Igor Szöke, Michal Fapšo, Martin Karafiát, Lukáš Burget, František Grézl, Petr Schwarz, Ondřej Glembek, Pavel Matějka, Jiří Kopecký, and Jan Černocký. Spoken term detection system based on combination of LVCSR and phonetic search. In *Machine Learning for Multimodal Interaction*, volume 4892/2008 of *Lecture Notes in Computer Science*, pages 237–247. Springer Berlin / Heidelberg, 2008c.

Paul Taylor. Hidden Markov models for grapheme to phoneme conversion. In *Proc. Interspeech'05*, pages 1973–1976, Lisbon, Portugal, September 2005.

Javier Tejedor, Dong Wang, Joe Frankel, Simon King, and José Colás. A comparison of grapheme and phoneme-based units for Spanish spoken term detection. *Speech Communication*, 50(11-12):980–991, November 2008. doi: 10.1016/j.specom.2008.03.005.

Javier Tejedor, Dong Wang, Simon King, Joe Frankel, and José Colás. Term-dependent confidence for out-of-vocabulary term detection. In *Proc. Interspeech'09*, pages 2131–2134, Brighton, UK, September 2009.

Kishan Thambiratmann and Sridha Sridharan. Rapid yet accurate speech indexing using dynamic match lattice spotting. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):346–357, January 2007.

Kishan Thambiratnam and Sridha Sridharan. Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting. In *Proc. ICASSP'05*, volume 1, pages 465–468, Philadelphia, Pennsylvania, USA, March 2005.

Jean-Manuel Van Thong, Pedro J. Moreno, Beth Logan, Blair Fidler, Katrina Maffey, and Matthew Moores. Speechbot: An experimental speech-based search engine for multimedia content on the web. *IEEE transactions on multimedia*, 4(1):88–96, March 2002.

Edward L. Thorndike and Irving Lorge. *The Teacher's Word Book of 30,000 words*. Teachers College, Columbia University, NY, 1944.

Kari Torkkola. An efficient way to learn English grapheme-to-phoneme rules automatically. In *Proc. ICASSP'93*, pages 199–202, Minneapolis, MN, USA, April 1993.

Ville T. Turunen and Mikko Kurimo. Using latent semantic indexing for morph-based spoken document retrieval. In *Proc. ICSLP'06*, pages 341–344, Pittsburgh, USA, September 2006.

Luboš Šmídl and Josef V. Psutka. Comparison of keyword spotting methods for searching in speech. In *Proc. ICSLP'06*, pages 1894–1897, Pittsburgh, USA, September 2006.

Dimitra Vergyri, Andreas Stolcke, Ramana Rao Gadde, and Wen Wang. The SRI 2006 spoken term detection system. In *Proc. NIST spoken term detection workshop (STD 2006)*, Gaithersburg, Maryland, USA, December 2006.

Dimitra Vergyri, Izhak Shafran, Andreas Stolcke, Ramana R. Gadde, Murat Akbacak, Brian Roark, and Wen Wang. The SRI/OGI 2006 spoken term detection system. In *Proc. Interspeech'07*, pages 2393–2396, Antwerp, Belgium, August 2007.

Paul Vozila, Jeff Adams, Yuliya Lobacheva, and Ryan Thomas. Grapheme to phoneme conversion and dictionary verification using graphonemes. In *Proc. Eurospeech'03*, pages 2469–2472, Geneva, Switzerland, September 2003.

Howard D. Wactlar, Alexander G. Hauptmann, and Michael J. Witbrock. Informedia: News-on-demand experiments in speech recognition. In *Proc. DARPA Speech Recognition Workshop 1996*, Harriman, NY, February 1996.

Alexander Waibel and Kai-Fu Lee, editors. *Readings in Speech recognition*. Morgan Kaufmann Publishers Inc, 1990.

Roy Wallace, Robbie Vogt, and Sridha Sridharan. A phonetic search approach to the 2006 NIST spoken term detection evaluation. In *Proc. Interspeech'07*, pages 2385–2388, Antwerp, Belgium, August 2007.

Roy Wallace, Robbie Vogt, and Sridha Sridharan. spoken term detection using fast phonetic decoding. In *Proc. ICASSP'09*, pages 4881–4884, Taipei, Taiwan, April 2009.

Dong Wang. Data resources for PhD thesis. Technical report, CSTR, University of Edinburgh, 2009. URL http://data.cstr.ed.ac.uk/dwang2/thesis-res.html.

Dong Wang, Joe Frankel, Tejedor Tejedor, and Simon King. A comparison of phone and grapheme-based spoken term detection. In *Proc. ICASSP'08*, pages 4969–4972, March 2008a. doi: 10.1109/ICASSP.2008.4518773.

Dong Wang, Ivan Himawan, Joe Frankel, and Simon King. A posterior approach for microphone array based speech recognition. In *Proc. Interspeech'08*, pages 996–999, September 2008b.

Dong Wang, Simon King, and Joe Frankel. Stochastic pronunciation modelling for spoken term detection. In *Proc. Interspeech'09*, pages 2135–2138, Brighton, UK, September 2009a.

Dong Wang, Simon King, Joe Frankel, and Peter Bell. Term-dependent confidence for out-of-vocabulary term detection. In *Proc. Interspeech'09*, pages 2139–2142, Brighton, UK, September 2009b.

Dong Wang, Tejedor Tejedor, Joe Frankel, and Simon King. Posterior-based confidence measures for spoken term detection. In *Proc. ICASSP'09*, pages 4889–4892, Taiwan, April 2009c.

Don Watson. *Death Sentence, The Decay of Public Language*. Knopf, Sydney, 2003.

Martin Wechsler and Peter Schäuble. Speech retrieval based on automatic indexing. In *Proc. Workshop on Multimedia Information Retrieval (MIRO'95)*, Glasgow, UK, September 1995.

Martin Wechsler, Eugen Munteanu, and Peter Schäuble. New techniques for open-vocabulary spoken document retrieval. In *Proc. ACM SIGIR 1998*, pages 20–27, Melbourne, Australia, August 1998.

Mitch Weintraub, Francoise Beaufays, Ze'ev Rivlin, Yochai Konig, and Andreas Stolcke. Neural-network based measures of confidence for word recognition. In *Proc. ICASSP'97*, pages 887–890, Munich, Bavaria, Germany, April 1997.

Mitchel Weintraub. LVCSR log-likelihood ratio scoring for keyword spotting. In *Proc. ICASSP'95*, volume 1, pages 297–300, Detroit, Michigan, USA, May 1995.

Frank Wessel, Klaus Macherey, and Ralf Schlüter. Using word probabilities as confidence measures. In *Proc. ICASSP'98*, volume 1, pages 225–228, Seattle, Washington, USA, May 1998.

Frank Wessel, Klaus Macherey, and Hermann Ney. A comparison of word graph and n-best list based confidence measures. In *Proc. Eurospeech'99*, pages 315–318, Budapest, Hungary, September 1999.

Frank Wessel, Ralf Schlter, Klaus Macherey, and Hermann Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, March 2001.

Lynn D. Wilcox and Marcia A. Bush. Training and search algorithms for an interactive wordspotting system. In *Proc. ICASSP'92*, volume 2, pages 97–100, San Francisco, CA, USA, March 1992.

Gethin Williams and Steve Renals. Confidence measures from local posterior probability estimates. *Computer Speech and Language*, 13(4):395–411, 1999.

Jay G. Wilpon, Chin-Hui Lee, and Lawrence R. Rabiner. Application of hidden Markov models for recognition of a limited set of words in unconstrained speech. In *Proc. ICASSP'89*, pages 254–257, Glasgow, UK, May 1989.

Jay. G. Wilpon, Lawrence R. Rabiner, Chin-Hui Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(11):1870–1878, 1990.

Michael J. Witbrock and Alexander G. Hauptmann. Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents. In *Proc. 2nd ACM International conference on Digital Libraries*, pages 30–35, Philadelphia PA, USA, 1997.

Martin Wöllmer, Florian Eyben, Joseph Keshet, Alex Graves, Björn Schuller, and Gerhard Rigoll. Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks. In *Proc. ICASSP'09*, pages 3949–3952, Taipei, Taiwan, April 2009.

Maria Wolters. A diphone-based text-to-speech system for Scottish gaelic. Master's thesis, University of Bonn, January 1997.

Maria Wolters and Antal van den Bosch. Automatic phonetic transcription of words based on sparse data. In *Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks*, pages 61–70, Prague, Czech Republic, April 1997.

Gunnar Evermann Phil Woodland. Large vocabulary decoding and confidence estimation using word posterior probabilities. In *Proc. ICASSP'2000*, 2000.

Phil Woodland, Sue E. Johnson, Pierre Jourlin, and Karen Spärck Jones. Effects of out of vocabulary words in spoken document retrieval. In *Proc. ACM SIGIR 2000*, pages 372–374, Athens, Greece, July 2000.

Hirofumi Yamamoto, Genichiro Kikui, Satoshi Nakamura, and Yoshinori Sagisaka. Speech recognition of foreign out-of-vocabulary words using a hierarchical language model. In *Proc. ICSLP'06*, pages 1870–1873, Pittsburgh, USA, September 2006.

Ali Yazgan and Murat Saraclar. Hybrid language models for out of vocabulary word detection in large vocabulary conversational speech recognition. In *Proc. ICASSP'04*, volume 1, pages 745–748, Montreal, Quebec, Canada, May 2004.

Sheryl R. Young. Detecting misrecognitions and out-of-vocabulary words. In *Proc. ICASSP'94*, volume 2, pages 21–24, Adelaide, SA, Australia, April 1994.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Engineering Department, Cambridge University, March 2006.

Steve J. Young, Martin Brown, Jonathan Trumbull Foote, Gareth J. F. Jones, and Karen Spärck Jones. Acoustic indexing for multimedia retrieval and browsing. In *Proc. ICASSP'97*, volume 1, pages 199–202, Munich, Bavaria, Germany, April 1997.

Peng Yu and Frank Seide. A hybrid word / phoneme-based approach for improved vocabulary-independent search in spontaneous speech. In *Proc. ICSLP'04*, pages 293–296, Jeju, Korea, October 2004.

Peng Yu and Frank Seide. Fast two-stage vocabulary independent search in spontaneous speech. In *Proc. ICASSP'05*, pages 481–484, Philadelphia, Pennsylvania, USA, March 2005.

Peng Yu, Kaijiang Chen, Chengyuan Ma, and Frank Seide. Vocabulary-independent indexing of spontaneous speech. *IEEE Transactions on Speech and Audio Processing*, 13(5):635–643, September 2005.

Chi Zhang, Ji Wu, Xi Xiao, and Zuoying Wang. Pronunciation variation modeling for Mandarin with accent. In *Proc. ICSLP'06*, pages 709–712, Pittsburgh, USA, September 2006.

Rong Zhang and Alexander I. Rudnicky. Word level confidence annotation using combinations of features. In *Proc. Eurospeech'01*, pages 2105–2108, Aalborg, Denmark, September 2001.