Early Stage Detection
of
Speech Recognition Errors
by

Stephen Choularton,
B.Sc.(Hons.)

This thesis, presented in fulfillment of the requirements for the Degree of Doctor of Philosophy, COMPSC, entitled:

Early Stage Detection of Speech Recognition Errors

written by **Stephen Choularton**

has been approved for the Division of Information and Communication Sciences Department of Computing and

was directed by Professor Robert Dale and Dr Steve Cassidy.

Professor Robert Dale

Dr Steve Cassidy

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Abstract

Machines mishear human utterances. This mishearing represents a gateway problem for speech applications, introducing errors into dialogues, or giving rise to clarification sub-dialogues that often cause as many problems as they solve.

Misrecognition is so ubiquitous that commercial speech recognizers make use of a confidence metric to deliver a numerical assessment of the probability that an utterance has been correctly heard. Each recognition is then classified as either correct or false depending on whether its confidence exceeds a pre-determined threshold. Even with an optimally chosen threshold value, this classification decision is still incorrect between 10 and 25% of the time.

The aim of this thesis is to improve upon this capability of the machine to know when it has misheard. We do this by first exploring techniques for assessing the likelihood of error separately in the acoustic domain and the language domain, and then combining these methods in a unified classification mechanism.

In the acoustic domain, we establish that individual speaker characteristics are a major factor in determining whether or not a speech recognizer will mishear an utterance, and that, using logistic regression, we can train a model to identify such speakers. Working with data that we know to be free of errors in the language domain, we show that we can identify the utterances of problematic speakers with 85% accuracy.

In the language domain, we explore a range of techniques for identifying out-of-language utterances, and determine the circumstances under which these different approaches are most appropriate. Working on data that we know to be free of errors in the acoustic domain, we show that a domain-independent technique can identify out-of-language errors with an accuracy of 82%.

Finally, we combine these techniques to provide a unified mechanism for predicting the recognizability of an utterance. Using a confirmatory dialogue turn as an example, we demonstrate that we can achieve an accuracy of over 95%.

Acknowledgements

It is traditional to acknowledge the people and institutions that played a significant role in enabling the completion of a Ph.D. In my case this represents quite a long list. I left school at 16, and never matriculated, so first I must thank Macquarie University for being brave enough to introduce the Jubilee Scheme allowing mature students to undertake degree studies without any formal entrance requirements. This allowed me to commence a Bachelor of Science Degree as I approached my 50th birthday.

I must thank Carolyn Kennett, the Director of the Numeracy Room Centre at Macquarie. The worst result I ever got was C for MATH101 but it was my sweetest! It would have been an F without Carolyn. I would commend the numeracy room to anyone. All the other members of staff fellow and fellow students were always helpful.

I would also like to thank Giorgio Martignoni. George was a fellow mature undergraduate. Quiet and unassuming, he would probably deny he was any assistance, but without his help I don't suppose I would have got to grips with computers at all. It has been an extra bonus to have made a new school friend at my age.

I must acknowledge and thank my principal supervisor, Robert Dale, who I had the good fortune to meet while undertaking a language technology unit in my second year. When you have no academic background yourself, the interest and support of someone like Robert makes all the difference. Without it, I have no doubt I would never have completed first an honors degree and then this Ph.D.

I must thank my associate supervisor, Steve Cassidy. As it turned out much of my work took a statistical turn and I found myself analyzing and processing sound files. It was my good fortune that these are areas where Steve's expertise and advice could often remove obstacles to my progress.

I must thank the Australian Government for awarding me an Australian Post Graduate Award and Macquarie for topping it up. Sadly everyone needs money to live.

Finally, I must thank my wife, Elizabeth Choularton, without whom none of this would have been possible or, indeed, worthwhile.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What this Thesis is About

Machines mishear human utterances. This mishearing represents a gateway problem for speech applications, introducing errors into dialogues, or giving rise to clarification sub-dialogues that often cause as many problems as they solve.

Misrecognition is so ubiquitous that commercial speech recognizers make use of a confidence metric to deliver a numerical assessment of the probability that an utterance has been correctly heard. Each recognition is then classified as either correct or false depending on whether its confidence exceeds a pre-determined threshold. Even with an optimally chosen threshold value, this classification decision is still incorrect between 10 and 25% of the time.

With these levels of recognition accuracy[1] an unacceptably large proportion of turns in dialogues are affected by errors in recognition of the sort shown in Examples 1.1 and 1.2 from the Carnegie Mellon Communicator Corpus [Bennet and Rudnicky, 2002].

(1.1) Said: something less than a hundred dollars a night
Heard: i something less than a two dollars a night

(1.2) Said: yes to manhattan new york
Heard: yes too it in newark

Errors arise from two causes:[2]

[1] We use the word *accuracy* when referring both to the proportion of hypotheses offered by a speech engine that are correct and the proportion of utterances our own predictive models classify as likely to be recognized correctly. In order to avoid confusion, we will use the term *recognition accuracy* to refer to the former and the term *prediction accuracy* to refer to the latter.

[2] Of course, because modern ASR systems are discrete and statistical in nature, some utterances, although within the acoustic and linguistic bounds of those experienced and expected by the machine (thus 'contained' in its model), will be inherently confusable. In consequence there is some upper limit

- the sound arriving at the machine is different from the sound expected by the machine, either because of variability in the way people pronounce words or the presence of side noise; and/or

- people say things using language that is unexpected by the machine.

The aim of this thesis is to improve upon the capability of the machine to know when it has misheard.[3] We do this by first exploring techniques for assessing the likelihood of error separately in the acoustic domain and the language domain, and then combining these methods in a unified classification mechanism.[4]

In the acoustic domain, we establish that individual speaker characteristics are a major factor in determining whether or not a speech recognizer will mishear an utterance, and that, using logistic regression, we can train a model to identify such speakers. Working with data that we know to be free of errors in the language domain, we show that we can identify the utterances of problematic speakers with 85% prediction accuracy.

In the language domain, we explore a range of techniques for identifying out-of-language utterances, and determine the circumstances under which these different approaches are most appropriate. Working on data that we know to be free of acoustic errors, we show that a domain-independent technique can identify out-of-language errors with a prediction accuracy of 82%.

Finally, we combine these techniques to provide a unified mechanism for predicting the recognizability of an utterance.[5] Using a dialogue state where the user confirms the system's understanding as an example, we demonstrate that, while the highest prediction accuracy achievable via an existing confidence measure is 91%, we can achieve a prediction accuracy of over 95%.

The remainder of this chapter elaborates on our starting points in this work. Section 1.2 looks at the current state-of-the-art in automatic speech recognition and why errors

to utterance recognition accuracy below 100%, and one has to recognize this in any attempt to improve error rates without actually changing or improving the recognizer itself.

[3]The particular focus of this thesis on the mishearing problem still leaves much work to be done in otherwise improving systems so speakers will be recognized better. Such steps extend from good dialogue design [Balentine and Morgan, 1999] one end of the spectrum to the use of active logic to detect and repair misrecognitions [Purang, 2001] at the other end of the spectrum. (See Section 2.2 for a review placing these matters in context.)

[4]Most fielded speech based systems are grammar based at this time and we develop our research with them particularly in mind.

[5]The techniques developed are recognizer-specific, that is they rely upon training on a particular recognizer using a particular set of acoustic and language models and are applicable to that recognizer, configured in that way. Owing to the different characteristics of different recognizers we are unable to offer a general solution that works for every recognizer; rather a technique that can be applied for a given recognizer to work out what might be misrecognized. The techniques were successfully used with three different recognizers: Sphinx4, SONIC and Nuance 8.

occur.

Nowadays, work such as this employs a range of software tools. Section 1.3 describes *Praat*, the tool we used to extract acoustic features, *Sphinx4*, the speech recognizer we used, and *R*, the statistical package we used. In addition, this sort of work often makes extensive use of corpora as data for empirical study. We describe our two main corpora: *TIDIGITS*, a well-known set of studio recorded utterances composed of digits, and the *CU Corpus* from an over-the-phone air travel information system. In addition we used the *Pizza Corpus*, from a fielded telephony based pizza ordering system, to study the rate of introduction of new words in dialogues; and *WSJ0*, derived from articles in the Wall Street Journal, to produce a language model.

Finally, in Section 1.4 we explain how we organize the thesis into five further chapters covering related work, the acoustic domain, the language domain, developing and evaluating various classifiers and some conclusions.

1.2 The Current State of the Art in Speech Recognition

As a starting point we will look at the current state of the art in speech recognition and then explore the causes of recognition errors. This provides a grounding for the balance of this thesis.

1.2.1 Automatic Speech Recognition

Spoken dialogue systems are prone to failure during use. Sometimes a system enters into a long running loop as nonrecognition is followed by repeated user attempts to be understood, as shown in Example 1.3 from the SpeechActs Corpus [Levow, 1999].[6]

(1.3) U: Give me the price for AT&T.
S: Hewlett-Packard is at 83 3/8, up 2 3/8 since yesterday.
U: Give me the price for AT&T.
S: Data General is at 63 1/4, up 1/3.
U: Give me the price for AT&T.
S: Hewlett-Packard is at 83 3/8, up 2 3/8 since yesterday.
U: Give me the price for American Telephone and Telegraph.
S: Excuse me?

To a significant extent the problem here is due to the nature of automatic speech recognition. It is possible to process the sounds that arrive at a microphone to bring out and identify certain features that can be used to try and identify which phoneme is being received. This process is not exact. The transition from one phoneme to another

[6] In the example U is the human user and S the system.

involves subtle graduated changes in the sound being produced and, of course, there are considerable variations in the way different people say the same phoneme. However, statistical models of the features are collected and used to decide what is probably being heard.

The recognition of phonemes in itself is not accurate enough to establish what has been said. Example 1.4 [from Williams, 1999, page 57] shows the phonemes which make up *bedouin* and *better one*.[7]

(1.4) a. bedouin => bcl b eh dx axr w ih n

b. better one => bcl b eh dx axr w ah n

As can be seen, the two very different utterances are all too easy to confuse, with only one phoneme being different (seventh phoneme, *ih* becomes *ah*). To assist the speech recognizer it is provided with a *language model* for additional guidance. A language model provides a prediction of the words that might be heard at any given point. The words proposed by the acoustic analysis are constrained by the language model and, ultimately, the recognized string that is offered maximizes the probability of fit to both the acoustic and language models. The language model can either be based upon an explicit grammar,[8] or upon an n-gram based language model.[9] N-gram language models should be able to provide very large (open vocabulary) recognition systems, but in practice, they appear to succeed best in the domain of the original corpus from which they were derived.

There are three main kinds of deployed automatic speech recognition tasks:

Large vocabulary speaker independent automatic speech recognizers such as used in the Switchboard automatic speech recognition task[10] (with systems from AT&T, BBN, Cambridge University, Dragon Systems, Johns Hopkins University, Mississippi State University, SRI International, and the University of Washington) [Greenberg et al., 2000]. These types of systems are used for tasks involving the transcription of large amounts of speech from diverse speakers. A typical application would be the transcription of radio news programs.

Small vocabulary speaker independent automatic speech recognizers such as the range of commercial products by Nuance and others. These types of systems are used in most of the telephony based spoken dialogue systems fielded commercially today, from banking to take-away food applications.

[7] The transcription scheme used here was the one developed for the TIMIT Acoustic-Phonetic Continuous Speech Corpus; see `http://www.ldc.upenn.edu/Catalog/readme_files/timit.readme.html`.

[8] An explicit grammar is a rule based definition of the syntax and vocabulary for what can be said.

[9] An n-gram based language model specifies the probability of one word following one or more other words, and is often extracted automatically from a corpus.

[10] This task involved the recognition of a large corpus of recorded phone conversations.

Large vocabulary speaker dependent automatic speech recognizers such as in the National Research Council of Canada's Bell Helicopter systems, the UK's Centrifuge and Tornado systems, and NASA systems.[11] These types of systems are often used to support a human such as a fighter pilot. In addition, in this group we might also place the commercial automatic speech recognizers used in such well known dictation systems as Dragon NaturallySpeaking and Microsoft's speech-enabled version of Windows.

Each of these tasks uses the same basic approach of maximizing the probability of fit of an utterance to both an acoustic model and a language model, although the commercial systems favor grammar-based, rather than n-gram based, language models. Ultimately automatic speech recognition is a task based on categorical perception. Any such system is capable of incorrect categorization, so spoken dialogue systems, and humans, often have to handle incorrect recognitions arising from this process. This phenomenon is ubiquitous:

- Word error rates[12] for the large vocabulary speaker independent automatic speech recognizers (such as for the Switchboard automatic speech recognition task mentioned above) run at up to 40% [Greenberg et al., 2000]. This is against the background of recognizing the utterances of many different speakers using over-the-phone speech.

- Small vocabulary speaker independent systems are often commercial implementations; error rates are often the subject of advertising claims, and may not be reliable. However, the Pizza Corpus (further discussed in Section 1.3.2), which was recognized with a Nuance recognizer, evidenced a word error rate of nearly 12% on utterances received over the telephone.[13]

- Large vocabulary speaker dependent automatic speech recognizers are often experimented with in military and space applications (where one might expect the best currently available automatic speech recognition technology to be used), but even here, as one reported case shows,[14] isolated digits were recognized with 92.5% recognition accuracy, digit triplets 92.7%, and command phrases 95.7%. This could still result in a command being incorrectly recognized in one-in-twenty instances.

Thus, spoken dialogue systems are faced with acting on recognitions containing errors in a material number of turns.

[11] As reviewed at `http://www.nrc-cnrc.gc.ca/~swail/sp_rec.html`.

[12] In this case we are referring to instances when a word being proposed by the speech engine differs from the word uttered.

[13] This word error rate is derived from our own work on the Pizza Corpus.

[14] See `http://www.nrc-cnrc.gc.ca/~swail/sp_rec.html` for the Tornado statistics.

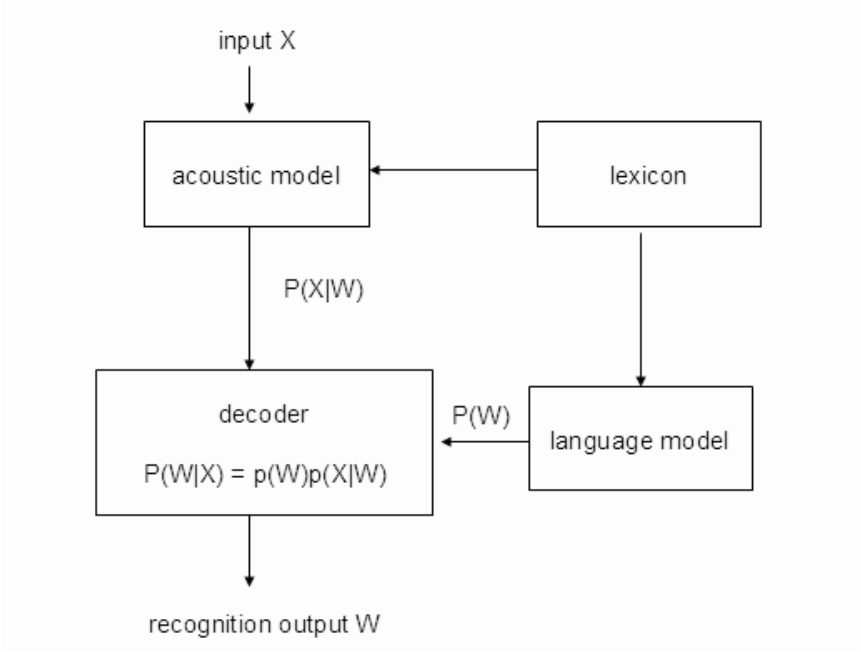

Figure 1.1: A Framework for statistical speech recognition [from Nanjo et al., 2000].

1.2.2 Why Errors Occur

The literature makes clear the reasons as to why automatic speech recognition errors occur. Speech engines produce hypotheses by seeking to find the best word sequence W for a given speech input X such that the combined score $P(W|X) = P(W) : P(X|W)$ is maximized (see Bahl et al. [1983] for an early reference to this approach). The framework is shown graphically in Figure 1.1. Ringger [2000, page 21] points out that this factors the recognition into two parts:

- a part dealing only with the acoustic aspects of speech; that is, the probability of this sound given the word sequence proposed, or $P(X|W)$) in the formula above; and
- a part dealing only with the word-level language; that is, the probability of the word sequence given the language model, or P(W) in the formula above.

Such a framework allows the taxonomization of the cause of errors into two parts:

Acoustic. The utterance 'heard' by the computer is dissimilar to the sounds modeled in the acoustic model.

Language. The utterance uses language not contained in the language model being employed by the speech engine.

Chase's Features	Our Category
Homophone substitution	Acoustic
Pronunciation problems in dictionary	Acoustic
Confused acoustic models	Acoustic
Out-of-vocabulary	Language
Language model overwhelming correct acoustics	Language
Search error	Mixed
Miscellaneous	Not applicable

Table 1.1: A more fine-grained taxonomy of the causes of errors [from Chase, 1997, Chapter 9].

Other more fine-grained taxonomies of the causes of error are possible: see, for example Chase [1997], who identified seven categories for the causes of error. Table 1.1 shows how Chase's error causes correspond to our simpler distinction; note that the finer-grained analysis is only possible post hoc on the basis of human-annotated data, and so is not appropriate in our situation, where we aim to pursue the classification of recognitions as they arise using only data available at the time of recognition.

Dissimilarity in the acoustic domain appears to arise along three dimensions:[15]

- Prosodic differences between a speaker's utterances, particularly along an axis from conversational to hyperarticulated speech, often driven by user perception of lack of understanding in the system [see, for example, Stifelman, 1993; Oviatt et al., 1998; Levow, 1998, 1999].
- Accent differences along an axis commencing with native dialect speakers, through 'standard' speakers, to non-native speakers importing accents from different languages.[16]
- Noise, such as a banging door, side-speech, or noise on a telephone line mingling with the speaker's utterance [see, for example, Rabiner and Juang, 1993, pages 305–309, for work on side noise].

Dissimilarity in the language model domain arises in two dimensions: deficiencies in vocabulary and/or syntax. These factors can be taxonomized as follows:

[15] In Section 3.6 we will refute the significance of the first factor, but it is commonly accepted as being significant in the literature.

[16] `http://lands.let.kun.nl/TSpublic/strik/pron-var/references.html` contains a lengthy bibliography of work that covers variations in pronunciation.

Unexpected On-focus Utterances: inadequacy of either the vocabulary or syntax to handle the scope of probable user utterances dealing with the focus of the dialogue at that time. For example, the focus of some particular point in a dialogue may be to obtain information about the age of a person, but the designer has not provided for the possible response, *My age is none of your business.*

Protest Utterances: inadequacy of either the vocabulary or syntax to handle the scope of probable user utterances when protesting errors introduced during previous foci. The inadequacy of a language model with respect to a fairly predictable protest *incorrect no* is clear in Example 1.5 from the Pizza Corpus.[17]

(1.5)
Prompt: What would you like for your third pizza?
Said: a large house @hes@ [fragment] ham and pine hawaiian
Heard: @hes@ large house supreme minus olives [confidence: 46]
Prompt: O.K. a large house supreme but no black olives
Said: incorrect no
Heard: [rejected] [confidence: 40]

Unexpected Off-focus Utterances: inadequacy of either the vocabulary or syntax to handle the scope of unexpected user utterances that are off-focus generally. This is particularly true of grammar-based systems where the language coverage tends to concentrate on the immediate focus of the dialogue. In consequence such things as requests to deal with some other item at any particular point in the dialogue (for example, to deal with car hire before booking a hotel, or obtain additional information before making a decision) will often fail to have been included in the language model.

Of course the sounds and the words that any system looks for are intimately linked with a pronunciation dictionary specifying the phonemes that may be considered to make up any word that can be included in the language model.

1.2.3 Summary

In this section we have seen that modern speech recognizers factor the recognition problem into two parts: one to do with acoustics and the other language. The acoustic model of the recognizer is used to produce suggested sets of phonemes; the language model of the recognizer is used to constrain these patterns in the interest of increased recognition accuracy. There are three main speech recognition tasks: large vocabulary speaker

[17] In the example *@hes@* indicates a hesitation by the speaker and the confidence score is shown by each recognition as [confidence: nn].

independent, small vocabulary speaker independent and large vocabulary speaker dependent. They all suffer from significant misrecognition rates with examples of word or utterance error rates of 40%, 12% and 5% respectively being shown.

The literature makes it clear that errors arise because the utterance 'heard' by the computer is dissimilar to the sounds modeled in the acoustic model, and/or the utterance uses language not contained in the language model being employed by the speech engine. This mirrors the same two dimensions, referred to in the previous paragraph, by which the speech recognition task is solved. In the acoustic domain, these differences can arise from prosodic and accent differences or side noise and noise on the line. In the language domain, there may be unexpected on-topic utterances, protest utterances or unexpected off-topic utterances.

1.3 Tools and Resources

It is common nowadays for work such as is reported on in this thesis to employ others' software tools. We were faced with the requirement for a number of such tools. Clearly we needed some tools to:

- extract acoustic features from sound files,
- perform recognitions, and
- carry out statistical studies.

They are reviewed in further detail in Section 1.3.1 below.

In addition one requires data resources. Much of the work in the field uses corpora for empirical study. We used several corpora. They came from a large set of spoken digits, an air travel information system, a pizza ordering system and a large corpus of read newspaper articles; these are reviewed in further detail in Section 1.3.2 below.

1.3.1 Software Employed

Praat

For the extraction of acoustic features we used Praat.[18] Praat is suited to batch and real time automatic extraction of features. The upper part of the screen shot shown in Figure 1.2 shows the waveform, with glottal pulses[19] superimposed, of the utterance *comp 123 practicals* automatically extracted by Praat.

[18] `http://www.fon.hum.uva.nl/praat/`.

[19] The glottal pulse is the period from the first sign of an increase in air flow associated with the glottis opening to its closing.

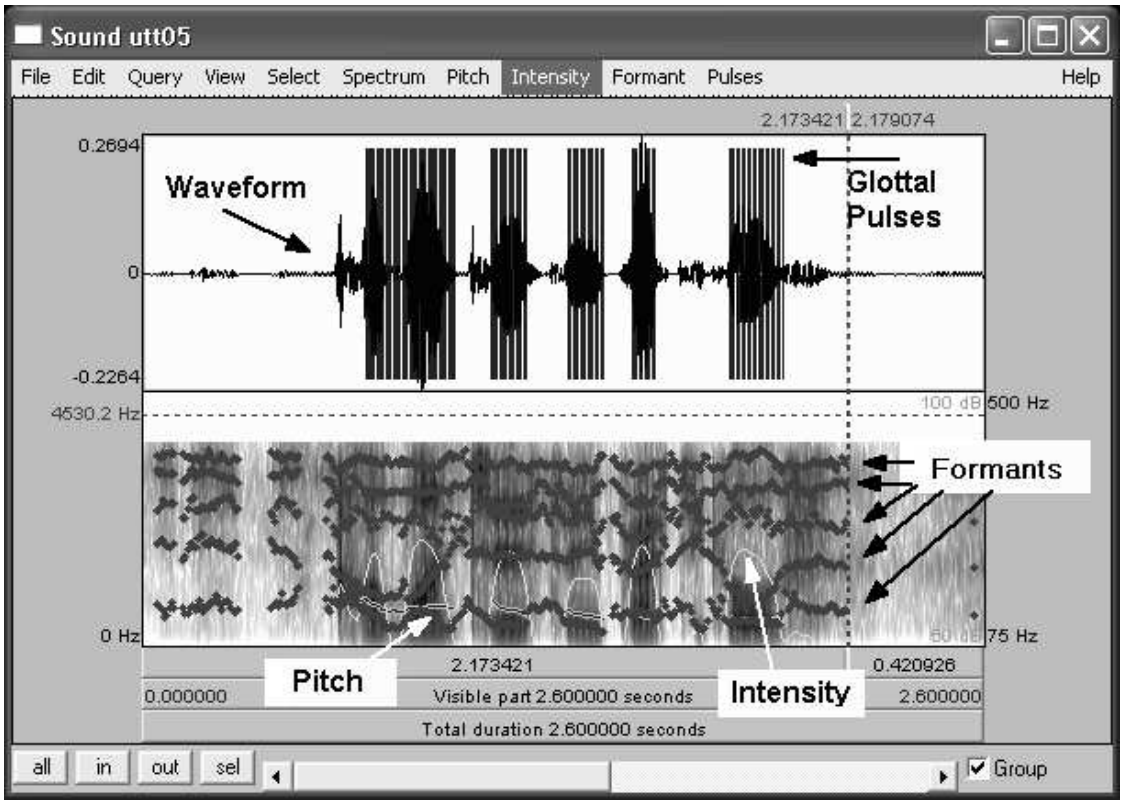

Figure 1.2: The prosodic features of the utterance *comp 123 practicals* automatically extracted by Praat.

Sphinx4

Sphinx4[20] is a state-of-the-art open-source speech recognition system written entirely in the $Java^{TM}$ programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT).

Use of an open source system had major advantages for us:

- Research systems are often set up to be able to process batch sets of utterances for experimental reasons. Sphinx4 is, and this was ideal for our experiments.
- Being open source, we could amend the code to produce the outputs we required.
- CMU supports Sphinx4 with a number of online user groups. This allows one to discuss design features with the designers. Again, with commercial systems these

[20] `http://cmusphinx.sourceforge.net/sphinx4/`.

are often closely-held commercial secrets.

In addition, it was possible to obtain both the acoustic training set and as well as the testing sets either directly, as they were shipped with Sphinx4 or from third parties such as the Linguistic Data Consortium.[21] Normally, the training sets are closely-held commercial assets.

R

R[22] provides statistical computing and graphics facilities. R is a free package, which is similar to the S language and environment. S was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues [R Development Core Team, 2006]. R provides a wide variety of statistical and graphical techniques, and is highly extensible. We found it invaluable for carrying out our logistic regression and other machine learning experiments.

1.3.2 Corpora

Throughout the course of this work we made heavy use of corpora. They provide an invaluable source of empirical evidence of phenomena. We used two corpora extensively through this thesis: TIDIGITS, a set of spoken digits that is well known in the speech field, and the CU Corpus produced by the University of Colorado. We also used two other corpora, the Pizza Corpus and WSJ0, to study the rate at which new words are introduced into dialogue and to build a language model respectively. The corpora are described below in the order in which we used them.

TIDIGITS

This is a US English dialectically balanced database consisting of more than 25,000 digit sequences spoken by over 300 men, women, and children.[23] The data was collected in a quiet environment and digitized at 20 kHz. The make-up of the speakers is shown in Table 1.2. The corpus was originally designed and collected at Texas Instruments, Inc. in 1980 and used initially in performance assessment tests of isolated-word speaker-dependent technology (see Doddington and Schalk [1981]).

The CU Corpus

In the late 1990s, the DARPA Communicator Program sponsored the development of a number of sophisticated spoken language dialog systems operating in the domain of

[21] http://www.ldc.upenn.edu/

[22] `http://www.r-project.org/`.

[23] This corpus has been published by the LDC; see `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S10`.

Gender	Number	Ages
Men	111	21–70
Women	114	17–59
Boys	50	6–14
Girls	51	8–15

Table 1.2: Speakers in the TIDIGITS Corpus.

air travel planning. As part of this program, in 2000, a major corpus of dialogues was collected from nine participating institutions [Walker et al., 2001].[24] One of the participating institutions was the University of Colorado, who have made their own data collection available for research; this is known as the CU Corpus. It contains the data they contributed to the DARPA project as well as additional data collected before and after that time. Example 1.6, while from the CMU Communicator Corpus, is a typical dialogue.

(1.6) S1. Hello. Welcome to the C M U Communicator. Are you a registered user?

U1. YES[25]

S2. What is your full name?

U2. DONNA PASSENGER

S3. We will use the standard profile. You may interrupt these instructions at any time by saying, good enough. The Communicator is a travel planning system with up to the minute flight information. It knows about major U.S. cities, and some international destinations. Here are some tips for a smooth interaction. Please speak clearly and naturally. Do not speak too quickly or too slowly. You can interrupt the system at any time by saying anything you wish. If you need to make a correction, just restate the new information. For example, if you'd like to depart from Cleveland instead of Pittsburgh, you can simply say, i'm departing from Cleveland. Now, here are some keywords you can use. To erase everything so far and start from the beginning, say "start over". To hear the last system response again, say "repeat". To end the call, say "good bye". This is the end of the instructions. If you need help at any time, please say,

[24]Part of this corpus is included in a larger corpus collected as part of a DARPA project. The larger corpus has been published by the LDC; see `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003S01` .

[25]The user's utterances are shown in capitals and the example above is as produced by the recognizer's log.

"help". Hello, donna passenger, I'm glad to hear from you again. You can ask me for help at any time. Where are you leaving from?

U3. I'M LEAVING FROM PITTSBURGH

S4. A flight from Pittsburgh. Where would you like to go?

U4. TO NEW YORK

S5. I know of three airports in New York. Did you want John F Kennedy, La Guardia, or Newark International?

U5. NEWARK INTERNATIONAL

S6. Traveling to Newark International. Departing on what day?

U6. MARCH THIRTY

S7. A flight tomorrow. And what time did you need to leave Pittsburgh?

U7. I WANNA LEAVE AT TWELVE O'CLOCK P M

The WSJ0 Corpus

During 1991, the DARPA Spoken Language Program initiated efforts to build a new corpus to support research on large-vocabulary continuous speech recognition systems. WSJ0 consists primarily of read speech with texts drawn from a machine-readable corpus of Wall Street Journal news text. Some spontaneous dictation is included in addition to the read speech.

The dictation portion was collected using journalists who dictated hypothetical news articles. Two microphones were used throughout: a close-talking Sennheiser HMD414 and a secondary microphone, which may vary. The corpora are thus offered in three configurations: the speech from the Sennheiser, the speech from the other microphone and the speech from both; all three sets include all transcriptions, tests, documentation, etc.

The vocabulary used in this corpus runs to some 129,247 words, which were listed in a Sphinx4 file in their alphabetical and phonemic forms. We used this corpus as the basis for our phoneme language modeling reported on in Section 4.4.

The Pizza Corpus

The Pizza Corpus arises from a trial of a pizza ordering system in Australia. A pizza chain fielded an operational telephony based spoken language system allowing customers to order pizzas for home delivery or pickup. The corpus is simply referred to as 'the Pizza Corpus'. Where necessary we have changed product names that might identify the business. Readers should particularly note that this system used a grammar for its language model, as is the case for most commercially deployed systems at this point in

time. Normally such systems have different grammars that are applied based upon the focus of the dialogue. A typical extract from the corpus is shown in Figure 1.3.

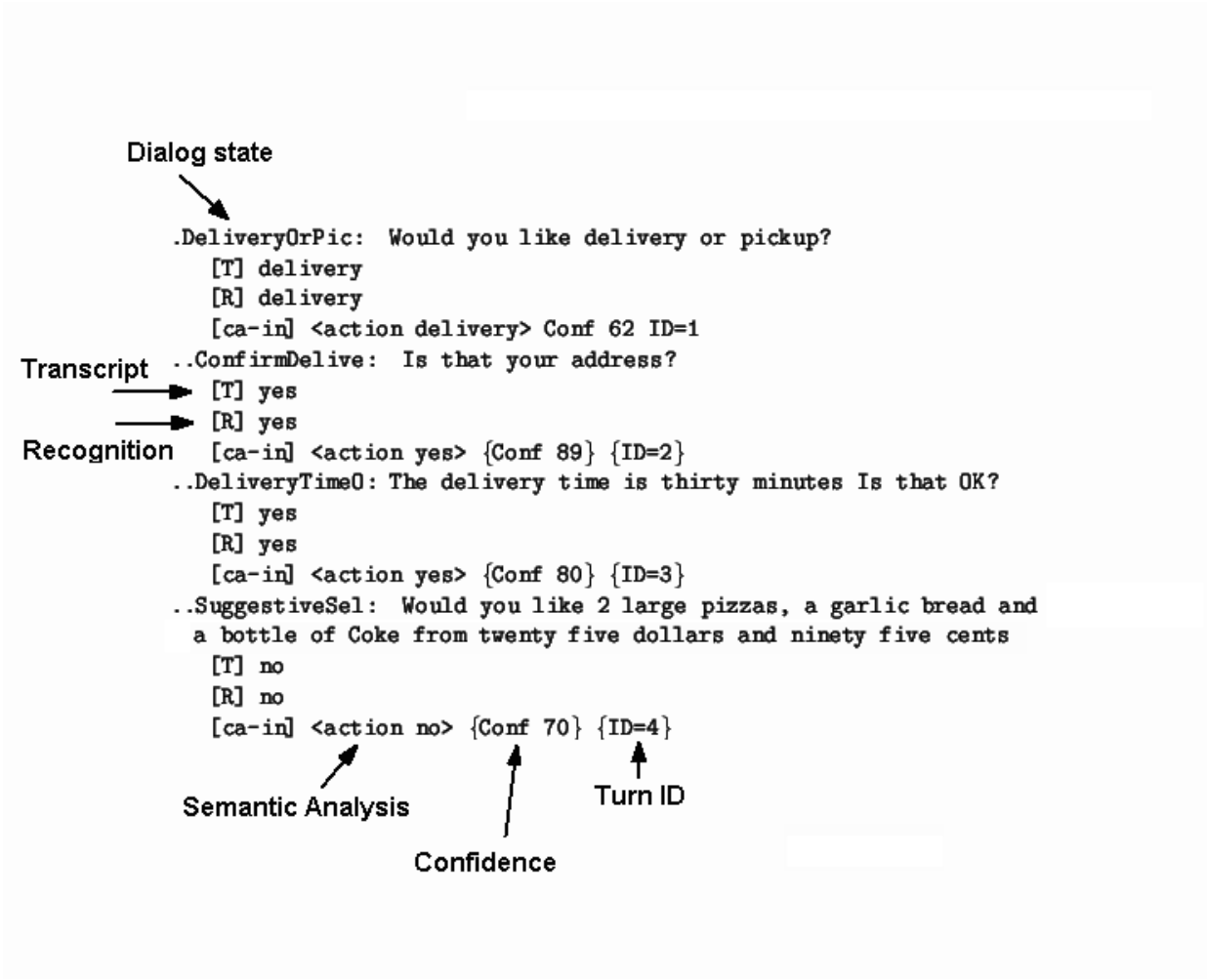

Figure 1.3: Excerpt from a Pizza Corpus dialogue.

Looking in detail at the first turn in the dialogue, *.DeliveryOrPic* specifies a dialogue state, indicating that this turn is to ascertain if the customer wishes to pick up his pizza from a local shop or have it delivered. The system prompt associated with this dialogue state, *Would you like delivery or pickup?*, is shown next. The recognition offered by the speech engine is shown two lines down (*[R] delivery*) with a human annotation shown one line above (*[T] delivery*). The next line provides the semantic analysis and the confidence estimate of the speech engine together with an ID number for the turn (*[ca-in] <action delivery> {Conf 62} {ID=1}*).

1.4 The Organization of this Thesis

In this chapter we identified a fundamental problem that faces spoken dialogue systems: speech recognition errors. We looked at the motivation for this work in the problems misrecognitions cause for deployed speech applications. We reviewed the background to the automatic speech recognition process and noted that these errors occur owing to a mismatch between the training set and the utterance being recognized. Finally we looked at the software we employed in our work and the corpora we used.

The balance of this thesis is divided into five chapters:

Chapter 2 – Related Work – where we review the literature relating to early stage detection of speech recognition errors. First we look at confidence, the current metric for identifying errors. We look at how it is used in modern recognizers, how it is calculated and how accurate it is as a predictor.

Speech recognizers use separate models of the acoustic features they expect to encounter and the language they expect to encounter. Errors occur when the recognizer encounters sounds or word that are not in these models. Throughout this thesis we pursue the causes of errors along these two dimensions: acoustic differences and language differences. We review the work done in the acoustic domain and then in the language domain. Then we identify outstanding research issues following from that work. Finally, we discuss why we have chosen to pursue our particular research issue: early stage detection of speech recognition errors.

Chapter 3 – Errors in the Acoustic Domain – where we explore the causes of errors in the acoustic domain. First we look at the general problem of modeling and classification. Then we explore our methods, the hypotheses we test and the materials we use. We use logistic regression as an analytic and predictive method. We test that we can predict errors and that they arise essentially from general characteristics of individual speakers, not from the different ways one speaker may render an utterance at different times.

We carry out our experiments first on the TIDIGITS Corpus then on the CU Corpus. Finally we explore the probable causes of errors in the acoustic domain before concluding. The chapter demonstrates, using data we know to be free of errors in the language domain, that one can identify the utterances of problematic speakers with 85% prediction accuracy using acoustic features alone and that errors are speaker rather than utterance related.

Chapter 4 – Errors in the Language Domain – where we explore the causes of errors in the language domain. First, we explore our methods, the hypothesis we test and the materials we use. Our basic approach is to compare a second

recognition using a more general language model with an original grammar based recognition to identify the presence of out-of-language words in the utterance. We use three techniques:

A Meta-word. Here we add a 'word' that models all patterns of phonemes to our grammar and see if it can identify words not in the original language model.

A Phoneme Language Model. Here we build a language model based purely on legitimate patterns of phonemes and see if a distance metric can identify words not in the original language model.

A Domain Language Model. Here we build a language model based purely on previous experience in the domain and see if it can identify words not in the original language model.

The chapter demonstrates, using data we know to be free of errors in the acoustic domain, that a domain independent technique can identify out-of-language errors with an prediction accuracy of 82%.

Chapter 5 – Experiments In Classification – where we bring together what we have learned in the preceding two chapters and use a range of machine learning techniques to produce classifications to predict errors. First, we explain our model building approach. We carried out experiments on TIDIGITS data containing both acoustic and language errors using logistic regression to form a baseline. Then we subjected the same data to experiments using six techniques:

- bagging,
- boosting,
- naive Bayes,
- neural networks,
- Random Forest,[26] and
- support vector machines.

Finally, we created a more real life data set from some 6,500 utterances in the CU Corpus that comprised user responses to the system's requests for clarifications. Having seeded the corpus with language errors, we subjected it to the same experiments. We used two recognizers: first Sphinx, and then, in order to be able to work with commercially produced confidence figures, Nuance 8. The chapter demonstrates that on real life data we can achieve an prediction accuracy of over 95%.

[26]Random Forest is a registered trademark.

Chapter 6 – Conclusion – where we review the thesis, discuss its place in the literature, identify further work and conclude.

Chapter 2

Related Work

2.1 Introduction

The published work relating to recognition errors and the problems they cause in dialogues is very wide. The fundamental problem flows from the fact that all too often the recognition being offered by a speech engine differs from the actual words in an utterance. These errors cascade into a dialogue in a manner that affects the very possibility of the dialogue being successfully concluded. Not only may the dialogue proceed on some misunderstanding, but, even if the error is picked up, clarification sub-dialogues with computers often prove difficult, leading to frustration and uncertainty in the user's mind. In this chapter we mainly restrict ourselves to early stage detection of errors (see Section 2.2) and ultimately we identify our objective: we hope to produce a highly accurate classifier of utterances; one that can reliably advise us if a recognition hypothesis is correct or incorrect. Such a technique would allow the computer to know accurately when it is facing a mishearing, mimicking a skill that humans appear to posses.

2.1.1 The Organization of this Chapter

Prior to concentrating solely on the early stage detection of speech recognition errors we undertook a wider review of errors in dialogue. Although it is not central to this thesis, we briefly present a summary of this material in Section 2.2, as it places early detection of errors in context as part of the larger problem of errors in dialogue.

In Section 2.3 we look at *confidence*. Confidence corresponds to a class of metrics currently generally produced by speech engines to determine the probability that a recognition is correct. This section looks at how the metric is used to manage dialogues, how it is calculated and how it performs if used to classify recognition hypotheses as correct or false. Given that the objective of this thesis is to produce a highly accurate classification of the correctness of recognitions, confidence represents the state of the art against which we must compare.

Speech recognizers, at least conceptually, tackle the problem of recognition by dividing it into two distinct areas: the aspects of how words sound and the aspects of what words are used in utterances. We chose to review the literature along these two dimensions:

- Section 2.4 looks in detail at the related work on errors in the acoustic domain. This work utilizes, to a greater or lesser extent, features such as pitch, formants, tempo and the like, extracted from acoustics. We build upon this work directly in Chapter 3.
- Section 2.5 looks in detail at the second dimension, related work on errors in the language domain. This covers handling unexpected words and syntax. This turns out to be a very different task to that of identifying acoustic errors. Productive approaches appear to be based upon using some more general language model for a second recognition of an utterance and then drawing conclusions, by comparing it with the original recognition or language model, as to whether or not one has encountered an out-of-language utterance. We build upon this work directly in Chapter 4.

Finally, Section 2.6 summarizes this review and identifies the research task we chose to pursue.

2.2 Early Stage Detection in Context

In this section we briefly present a summary of the wider work on errors in dialogue. This material is not central to this thesis, but it places early detection of errors in context as part of the larger problem of errors in dialogue.

McTear et al. [2005] suggest that handling errors has traditionally been looked at in terms of a number of stages:

- prevention, where one tries to encourage users to say things the recognizer will find easy to handle;
- detection, which itself falls into two sub areas: early stage detection, where one spots incorrect recognitions as they arise and which is the subject of this thesis, and late detection, where one spots the changes in the behavior of users when a incorrect fact or instruction has already entered a dialogue; and
- recovery, where one tries to repair the incorrect recognition by entering a clarification sub-dialogue or using some automatic technique to correct the recognition.

In this section we will look first at trying to get people to say things that the computer finds easy to recognize (Section 2.2.1), then we look at the steps taken to

cope with errors when they arise (Section 2.2.2). Finally, we look at the work on detection of errors (Section 2.2.3).

2.2.1 Prevention

There is literature on how to design spoken language systems in a manner that leads users to say things that are easy to recognize correctly and act upon; these extend from practitioners' books like Balentine and Morgan [1999], to academic papers like Zoltan-Ford [1991]. This type of activity may be referred to as *good dialogue design* and is to be commended. Two examples from Balentine and Morgan demonstrate this. The first shows a system dealing poorly with a user's silence then dealing effectively with it by encouraging them to use words in the systems language model.

(2.1) *System: Main Menu <beep>*
User: <two-second timeout without speech>
System: I didn't hear you. Please make a selection or say help.

(2.2) *System: Main Menu <beep>*
User: <two-second timeout without speech>
System: Please say one of the following: balance, quotes, purchases, help, Operator.
User: Quotes

The second example shows it is better to ask a direct question in order to obtain a spontaneous *yes* or *no* from a user. Example 2.3 shows questions to avoid and 2.4 the more direct form of question.

(2.3) *If that is correct, please say yes otherwise say no.*
Do you want me to play that message?... answer yes or no.
To place another order, say yes now.

(2.4) *Is that correct?*
Play message?
Do you want to place another order?

Of course, good dialogue practice will not prevent at least some recognition errors occurring.

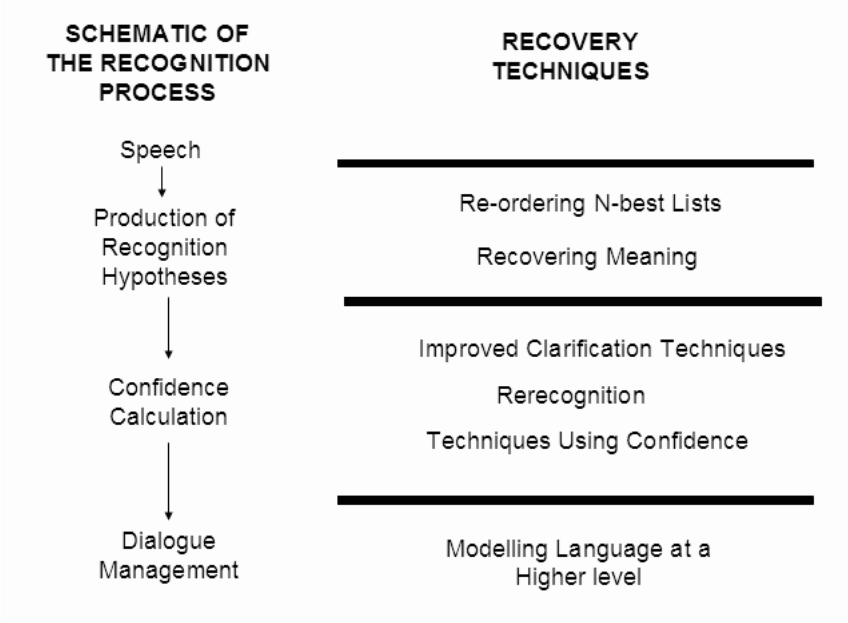

Figure 2.1: The place of recovery techniques in the dialogue process.

Another strand of literature addresses disfluencies. Disfluencies cover such phenomena as false starts, repetitions, and self-corrections, and are intra-utterance phenomena, such as:

(2.5) Can I fly ... uh drive to Hobart?

As such they are corrected by the speaker before the utterance is completed, and before any error is introduced into the dialogue. Oviatt [1995] indicates that disfluencies may be expected in less than one in a hundred utterances in cooperative dialogue.[1] She shows that design methods which guide users' speech into briefer sentences have the potential to eliminate the majority of spoken disfluencies.

2.2.2 Error Recovery

There is a very considerable literature on error recovery both at the stage an hypothesis is first offered and also once an error has entered a dialogue. Figure 2.1 shows how recovery techniques fit into the dialogue process. First are techniques that are used when the recognition is first produced and that work on n-best lists[2] or the most favored hypothesis:

[1] That is the sort of task oriented dialogues between a human and a computer that are the main domain for the work in this thesis (e.g. banking systems, ticketing systems and booking systems).

[2] Speech recognizers typically produce a number of hypotheses for any particular utterance. These are arranged as n-best lists, ranked by some metric as to their likelihood. The n refers to the length of the list.

Re-ordering N-best Lists. Chotimongkol [2001] uses a wide range of features from the automatic speech recognizer to train a classifier to pick an alternate hypothesis from the n-best list. In addition, non-statistical approaches to reordering these lists are used: Bousquet-Vernhettes and Vigouroux [2003] use semantic parsing, Zollo [2003] uses parsing into speech acts and Gurevych and Porzel [2003] use a domain ontology.

Lattice Rescoring. The many hypotheses contained in the n-best list can be compactly represented in the form of a word lattice. Various techniques can be used to rescore the lattice with a view to improving accuracy:

Statistical Parsers. Statistical parsers take into account dependencies in the hypothesis that are arbitrarily distant, such as the occurrence of the words *either* and *or*. Parsing models can capture these sorts of dependencies in a way n-gram models cannot. One example of this work is Roark [2002] where small, but distinct, improvements in word accuracy of just under 1% from around 89.2% to around the 90% level were achieved.

Multilayered Perceptrons. Neural networks can be used to estimate phoneme *a posteriori* probabilities and these new probabilities used to rescore the lattice. Pinto et al. [2007] found inconclusive results with the technique when used in keyword spotting experiments.

Articulatory Units. Neural networks can be used to estimate *a posteriori* probabilities for articulatory features such as whether a sound is a fricative or nasal, whether it is produced labially or glottally and so on. The lattice is re-written in terms of articulatory features and re-scored. Hacioglu et al. [2004] used this technique and improvements were seen but the levels were not stated in terms of word accuracy.

Recovering Meaning. Grishman [1998] tries to recover meaning based upon a model assuming a noisy channel corrupting the correct hypothesis through insertions, deletions and substitutions. Modeling substitution as a deletion followed by an insertion, he tries to modify the hypothesis to skip these events.

Correction. Ringger and Allen [1996a,b,1997] identify recognition errors that occur in a corpus where both the recognitions and annotations are available and then use the statistical mapping techniques that were developed for machine translation (see Brown et al. [1990]) to correct the initially offered hypothesis.

Next we have techniques that are used during or based upon the calculation of confidence:

Improved Confirmation Techniques. Skantze [2003, 2005] uses Wizard-of-Oz studies to investigate the strategies humans use when trying to clarify possible misunderstandings. McTear et al. [2005] develop strategies based upon the observation that system responses that do not involve signals of non-understanding appear to handle errors best. Bulyko et al. [2005] explore how the actual words used in a confirmation can affect the success of confirmations.

Rerecognition. Orlandi et al. [2003] modify their language model to exclude language contained in any hypotheses that appears to be incorrect owing to the user's response and subject the utterance to a second recognition.

Techniques using Confidence. Palmer and Ostendorf [2001] use confidence as a trigger to propose an alternate hypothesis when a named entity appears to be misrecognized; Horvitz and Paek [2001], Libuda [2001] and Prodanov and Drygajlo [2005] use confidence to determine the next action in a dialogue.

Finally we have techniques that are based upon an overview of the progress of the dialogue:

Modeling Language at a Higher Level. Many researchers try to model language at a higher level than words and grammars, in order to catch and repair errors. Allen [1995] models language at the level of speech acts to try and identify errors. McRoy and Hirst [1995] use abductive reasoning to catch errors at the level of speech act misunderstanding. Danieli [1996] uses dialogue expectations to model what the next utterance might be about, and to account for how the next utterance might be related to the last. Perlis and Purang [1996] try to use meta-dialogue and meta-reasoning to cope with mistakes. Purang [2001] tries to take this a step further using active logic as a general basis for agents to detect and repair their own mistakes.

There is work on the question of normalizing individual speakers that goes back some time (see for example Cohen [1995], Molau et al. [2000], Pitz et al. [2001] and Wellington et al. [2002]) and there has been a considerable amount of publication recently (see Öhgren [2007], Wang et al. [2008], Monaghan et al. [2008] and Saraswathi and Geetha [2008]). Much of this work concentrates on vocal tract normalization, which clearly improves recognition accuracy, but Saraswathi and Geetha show that time scale modification also improves recognition accuracy and Monaghan et al. also looks at glottal pulse rates. *Normal*, in this context, is defined as the characteristics of a standard speaker as represented in the acoustic models being used for recognition. Humans appear to have little difficulty in handling these variations [Smith et al., 2005].

2.2.3 Detection

If one cannot eliminate all errors, the importance of being able to detect when errors occur is clear. One may attempt to discern errors as they arise, that is, to know immediately that the recognition hypothesis differs from the utterance it purports to represent. This is referred to as *early stage detection of errors*; Sections 2.4 and 2.5 are devoted to the topic and we do not review it further in this section.

One may attempt to discern errors once they have entered into a dialogue. This is referred to as *late detection of errors* upon which there is a considerable literature. Much of it is based upon observing human behavior under error conditions (see for example Stifelman [1993], Krahmer et al. [2001], Shin et al. [2002], Bousquet-Vernhettes et al. [2003], Choularton and Dale [2004] and Rotaru and Litman [2006]). Essentially, the work shows that humans adopt principled patterns of behavior when faced with errors. Therefore, spotting those behaviors can be used to predict that one is in part of a dialogue where the system is proceeding on the basis of an error.

In addition, there is strand of literature that examines in greater depth the issues of understanding between speakers (see, for example, Grosz and Sidner [1990]; Traum [1994]). This looks at the grounding actions used by participants to confirm facts as they are introduced into a dialogue. Grosz and Sidner develop SharedPlan Theory to account for this phenomenon and Traum extends speech act theory [Austin, 1962; Searle, 1969, 1979] with the introduction of a *grounding* act. This work concentrates on deeper aspects of understanding in dialogue than simple mishearing and we do not pursue it here.

2.2.4 Summary

Recognition errors in dialogue are a major problem and have given rise to a considerable amount of research and literature. This thesis treats only one aspect of the problem, early detection of errors. However, in the pantheon of errors, this gateway problem is very significant. Equipping spoken dialogue systems with the ability to know if they have misheard or not will significantly improve the ability of subsequent error handling to work effectively.

2.3 Confidence

At the beginning of this thesis, we made the observation that confidence, as currently computed by speech recognizers, lacks the prediction accuracy required to adequately fulfil the role of deciding if an utterance has been misheard. In this section we look at how confidence is used in the field, how confidence is calculated and the problems in using confidence to accurately predict if any particular utterance is correctly recognized

or not.

In this thesis we talk about the prediction accuracy of various metrics in classifying outcomes. With respect to some methods, such as decision trees, prediction accuracy is a simple concept. The method predicts some outcomes and they are compared with a gold standard; the number of false positives and false negatives is ascertained and calculated as a percentage of the total number of predictions. This figure is the error rate and its complement can ordinarily be referred to as *accuracy*. Of course the same approach cannot be taken with a continuous metric such as confidence, or indeed the predictions produced by logistic regression or support vector machines. In order to grapple with the concept of prediction accuracy for these sorts of methods, we also need to take into account some breakpoint in the value of the continuous metric; this allows us to analyze it in the same way as a method that produces a binary outcome. Generally, the figure for prediction accuracy varies at various different breakpoints. Normally, we calculate an optimal value for prediction accuracy at the breakpoint that maximizes pairwise agreement between the predictions and outcomes.

In the next section we review how confidence is currently used.

2.3.1 How Confidence is Used

Confidence is highly indicative of the likelihood of error; metrics of confidence were developed to assist designers of applications to decide which confirmation strategy to adopt based upon how confident the system is in the correctness of the hypothesis being proposed. In cases where the confidence is high, the system will normally proceed on the assumption that the hypothesis is indeed correct. In situations where the system produced a recognition but is unsure as to its correctness (perhaps confidence is below 80%),[3] it is typical for a confirmatory question to be asked, as in Example 2.6. Clearly, in these cases, the system also has to allow for the user's protest when recognizing the next response, and to negotiate an appropriate correction of the error. The user might well start off by protesting with the utterance *no, no, no*, or *stupid machine*. If these are not allowed for by the grammar, further errors will occur.

(2.6) System: Where do you want to fly to?
User's response: Austin
System's recognition: Boston
System: OK, that's flying to Boston. Where are you departing from?

[3]The thresholds referred to in this section are indicative. In real systems, levels are chosen to produce a certain level of false positives or false negative based upon policy decisions. It might, for example, be very important to have a high degree of certainty about the sum of money a person wishes to transfer out of their bank account; in such a case a very high confidence threshold would be adopted as a hurdle below which confirmation would be sought.

User: No!
System: Sorry, I must have misheard, where did you want to fly to?

While Example 2.6 shows an *implicit* confirmation in the system's second utterance, Example 2.7 shows an *explicit* confirmation, normally reserved for those cases where there is greater doubt that the recognition is correct (perhaps confidence is below 65%):

(2.7) System: Where do you want to fly to?
User's response: Can I fly to Austin?
System's recognition: Can I fly to Boston?
System: Sorry, did you say Boston?

In cases where the confidence level is very low, perhaps below 45%, a typical strategy is for the system to reject even its best recognition, and to re-ask the question:

(2.8) System: Where do you want to fly to?
User's response: Can I fly to Austin?
System's recognition: (Rejected for want of confidence.)
System: Sorry, I didn't get that. Where do you want to fly to?

The levels of confidence at which these different tactics are adopted are often determined pragmatically by studies of the actual levels of errors experienced in fielded systems at particular dialogue states, and do not necessarily enjoy fixed levels across dialogue states or in different dialogues.

2.3.2 How Confidence is Calculated

As previously mentioned, confidence is a measure which endeavors to indicate how confident the speech engine is, normally at an utterance level, that it is delivering, in text, a true representation of what was said. The ability to determine that one has recognized correctly is, of course, a key performance factor, and the various commercial speech engines tend not to publish how they calculate confidence. However, Gillick et al. [1998], San-Segundo et al. [2000] and Zhang and Rudnicky [2001] report on the types of factors used to calculate this index in their own systems or more generally.

Zhang and Rudnicky point to a selection of the literature that considers confidence metrics based upon the automatic speech recognizer's outputs or processes. The literature considers the production of confidence for each word; the confidence for an entire utterance is calculated as the product of the confidence for each word in it. The four main groups of features use are based upon acoustic, language model, n-best list or word lattice features:

Acoustic Features. [Chase, 1997] These cover comparisons between the phones used in the ultimate word based recognition and the phones revealed by a phoneme

based recognition, and the ratio between the acoustic scores for the same two sets of data. The more similar the two sets of data, the greater the confidence in the recognition.

Language Model Features. [Carpenter et al., 2001] These are based upon using the back-off mode of an n-gram language model; owing to sparse data, often a trigram containing the word will not be available, so the system will resort to a bigram or a unigram. If something less than a full trigram is used, one is less confident in the result.

Word Lattice Features. [Wessel et al., 1998] These are based upon using the posterior word probability (from acoustic scores and/or language model scores) across the number of paths which lead to the word in question. That is, the higher the aggregate of the scores for the word in different hypotheses, the higher the confidence.

N-best List Features. [Chase, 1997] These compare the scores across all the paths containing a hypothesized word with scores across the full n-best list, or the total number of paths containing the word with the total number of paths in the n-best list. Confidence is higher if a word appears in many of the hypotheses offered, or with a higher aggregate score.

Turning to two particular recognizers, Gillick et al. [1998] report on the Dragon system, which uses a blend of six predictors for confidence:

- the duration of the word;
- the language model score for the word;
- the proportion of times the word appears in the appropriate location in the n-best list of hypotheses for the utterance (n = 100 in this system);
- the average over the period of time covered by the utterance of the word of the number of Hidden Markov Model states (across the whole vocabulary) active in that time;
- a normalized acoustic score for the word; and
- the log of the number of recognized words in the utterance.

San-Segundo et al. [2000] report on the calculation of confidence in the CU Communicator system. It uses three features:

Language model-back off. This feature gives more confidence to recognitions incorporating words in trigrams in the language model rather than bigrams or unigrams.

Language model score. This feature uses the log-probability for each word in the sequence (again as derived from the language model).

Phonetic length of the word. This feature is based on the observation that often, when an out-of-vocabulary word is encountered, the automatic speech recognizer will substitute a sequence of short, monosyllabic words.

As can be seen, all these approaches to the calculation of confidence involve a posterior use of outputs of the automatic speech recognizer to produce a metric that can be used to determine how much one can rely upon the hypothesis.

More recently, Bohus and Rudnicky [2006] describe a belief updating method that is used to modify the automatic speech recognizer's confidence metric and appears to have a significant effect on task success; and Higashinaka et al. [2006] incorporate 12 discourse features, based on Grice [1975], into confidence scores. In both cases a traditional confidence metric is initially used to assess if the semantic content of an utterance should be accepted, but the grounding of the facts it represents is reinforced or challenged by subsequent events in the dialogue. Machine learning algorithms train models based upon data comprised by these subsequent events.

Bohus and Rudnicky [2006] have their dialogue system maintain a set of belief hypotheses for each concept in their dialogue and update them after every turn. They use a wide range of features extending from confidence scores, through acoustic and prosodic features, to empirically determined confusability scores. They report a reduction in the error rate from 8.6% to 5.7%.

Higashinaka et al. [2006] adopt a similar framework but derive their features from the dialogue. Following Grice's maxims for co-operative dialogue of quantity, quality and manner they identified one, seven and four features for each maxim respectively. The feature relating to quality is, for example, a keyword pair count each time the system uses a keyword that the user also incorporates in their next utterance. Example 2.9 shows just such a case relating to a weather information system.

(2.9) System: "Are you interested in the weather in Tokyo?"
User: "The weather in Tokyo."

Here the keyword is *Tokyo*. Although this might seem counter-intuitive, they use a high count to indicate a problematic recognition. They feel that Grice's fourth maxim of relation was abided by, in any event, in task-oriented dialogues and did not take it into account. They improved recognition accuracy in identifying the correct concept value from an F-score of 0.685 when using only recognizer outputs to, one of 0.791 by incorporating their 'Gricean' features.

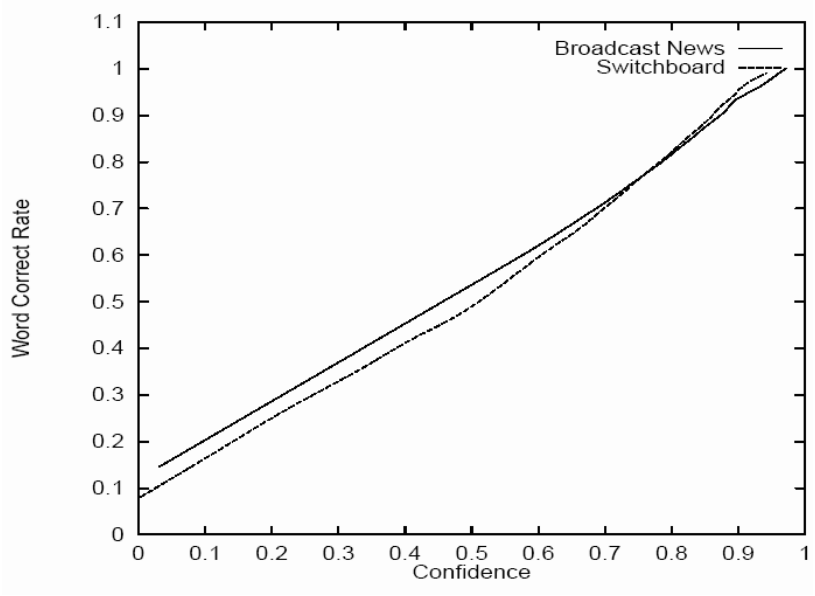

Figure 2.2: Word correctness vs confidence estimate [from Gillick et al., 1998].

2.3.3 Using Confidence for Predictions

Gillick et al. [1998] demonstrate the predictive qualities of their confidence index. They study both the Broadcast News Corpus[4] and the Switchboard Corpus.[5] Figure 2.2 shows the figures they produce for correctness as a function of confidence value. The x-axis shows the confidence values associated with a given word, while the y-axis shows the proportion of words correctly recognized at that confidence value. The graph shows, in their case, that confidence correlates well with the correctness of the hypothesis for the word.

Mengusoglu and Ris [2001] provide a graph for word correctness when working on the Phonebook Corpus [Pitrelli et al., 1995], a database of phonetically-rich isolated words. Reproduced here as Figure 2.3, it shows the same correlation between correctness and confidence, although this time the shape of the association is sigmoid, which might indicate that the method used to calculate confidence was less continuous than in Gillick et al.'s case.

However, while the hypothesis that confidence scores reflect the probability of lexical correctness is confirmed, this does not mean that confidence is a good classifier of correct and incorrect hypotheses. We have already discussed the concept of prediction accuracy in relation to such classifications in the introduction to this section. Mengusoglu and Ris [2001] report on this and we reproduce as Figure 2.4 a graph plotting the word

[4]This is a large corpus of broadcast news material - `http://www.ldc.upenn.edu/`.

[5]This is a large corpus of recorded phone conversations - `http://www.ldc.upenn.edu/`.

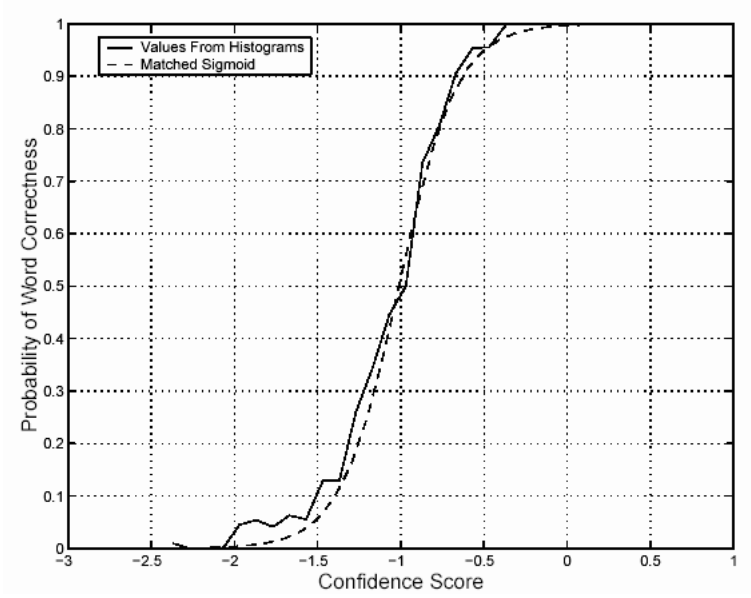

Figure 2.3: Probability of word correctness vs. confidence score [from Mengusoglu and Ris, 2001].

rejection rate, that is, the level of confidence at which one decides to reject or not, against the resulting classification error rates. As can be seen, the error rate is around 23% at its optimum, corresponding to an prediction accuracy of 77% in this case.

Skantze [2003] carried out a study of the prediction accuracy of confidence, using a Nuance recognizer. The best result came at a confidence figure of 40%, where prediction accuracy was 76.4%. Chen and Hasegawa-Johnson [2004] indicated that they produced a prediction accuracy rate significantly below 75% when using confidence to classify errors. In addition Gabsdil [2003] reported that when using confidence with 66% as the breakpoint, an F-score of only 63.6% was achieved. On the other hand Gurevych and Porzel [2003] reported 84% correctness with a train enquiry system.

It is always difficult to know if reported studies are strictly comparable, but the figures seem to indicate generally that confidence, when used with some optimal breakpoint, is classifying recognitions as correct or incorrect with a prediction accuracy of something like 75% with occasional outliers such as Gurevych and Porzel [2003] at 84%.

2.3.4 Review

Confidence is used in fielded spoken dialogue systems to determine if a dialogue can proceed on the current recognition or if, for want of confidence, the dialogue should enter a clarification sub-dialogue to confirm the fact(s) or instruction(s) contained in the recognition.

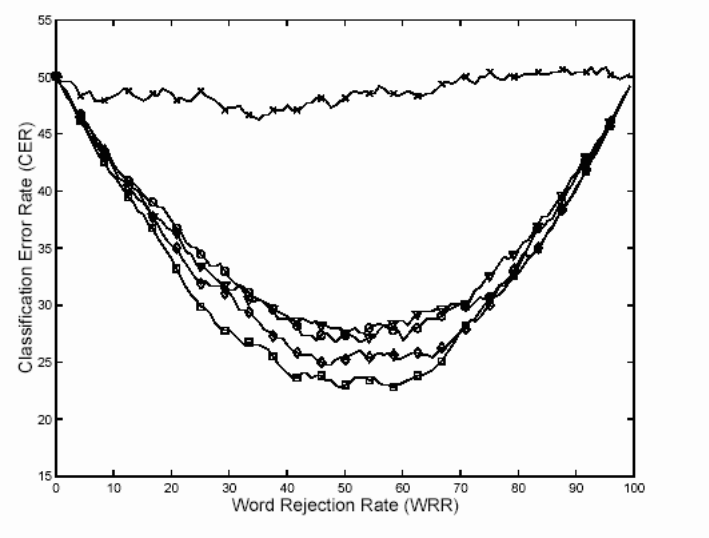

Figure 2.4: Word rejection rate vs. classification error rate [from Mengusoglu and Ris, 2001].

Confidence is calculated in various ways, but generally it relies on data arising solely within the speech engine such as the acoustic scores calculated during recognition.

Confidence correlates well with correctness, but when taken with a breakpoint turns out to be a poor classifier. The literature indicates that it often performs with little better than a 75% prediction accuracy.

In the next section we review the work relating to errors in the acoustic domain. This work explores the possibility of using many acoustic features lost or ignored in the recognition process, such as pitch, intensity, formants and so on, as a source of data to classify recognitions as correct or incorrect.

2.4 Errors Arising in the Acoustic Domain

2.4.1 Introduction

There is a considerable literature on the many causes of recognition errors that arise from influences in the acoustic domain. We briefly review this work in Section 2.4.2. Our principal concern is to be able to predict if errors will arise from an inspection of an utterance's acoustics. Therefore in Section 2.4.3 we first look in greater depth at the sorts of changes that appear to occur in speech at the times misrecognitions happen, and then at the use of acoustic features to classify misrecognized utterances. Generally, this work is motivated by the idea that hyperarticulation is heavily involved in the process of generating errors.

When users run into recognition problems with a system, or even when they perceive they might have problems speaking to a machine, they often stop speaking in a conversational style, and they speak more slowly and with greater emphasis; this behavior is called *hyperarticulation*, and largely mirrors the way people speak to children, the hard-of-hearing and non-native speakers showing difficulties with the language. As the corpora upon which speech recognizers are trained tend to be collections of conversational speech, hyperarticulated speech can be thought of as different and this difference should increase the chance of misrecognition. Indeed it is estimated that the chance of recognition errors doubles in the portion of the dialogue which immediately follows an error [Oviatt et al., 1998; Levow, 1998, 1999]. This view suggests that it is a change in prosodic features between one utterance and another by the same speaker that is a major contributor to the problem. However, there is a contrary view that we also explore: this view is that in some way the quality of how an individual speaks generally is more significant. In consequence, we complete our review of acoustic errors with an exploration of the work on *sheep* and *goats*, exploring problematic speakers [Doddington et al., 1998]. As we wished to pursue this view we felt we should include material in this section of the review that might normally be considered not to be strictly within the scope of early detection as it is associated with subsequent errors in recognition once the user has already been misrecognized.

2.4.2 Acoustic Errors Generally

Much of the work on identifying errors in the acoustic domain is based upon extracting acoustic features from an utterance, and relies heavily on hyperarticulation as the basis for the variations that cause errors (see, for example, Stifelman [1993]; Oviatt et al. [1998]; Levow [1998, 1999]). However, the literature shows that there are many potential sources of difference in the acoustic domain. Those who have tested a system trained on male speech with female speech or vice versa will know performance deteriorates. There is also literature dealing with the problems caused for recognition by many other types of speech and side-noise generally:

Pronunciation Variation. Not only do non-native speakers bring to a language their own pronunciation, but native speakers also often pronounce words in non-standard ways. President Bush's way of pronouncing *nuclear* more like *nukular* is one example.[6] There is a considerable literature looking at this problem[7] but it does not directly assist with the identification of utterances likely to be misrecognized.

[6] In fairness, many US Presidents have had problems with this word. President Eisenhower also pronounced the word *nukular* and President Carter *nu-kyir*. Most speakers pronounce it *NU-klee-er*. The phenomenon is called metathesis: the switching of two adjacent sounds within a word.

[7] See for example `http://lands.let.kun.nl/TSpublic/strik/pron-var/references.html` for a bibliography citing over 100 works.

Cold Speech. When speakers have a cold, recognition accuracy deteriorates (see for example Tull and Rutledge [1996]; Martin and Przybocki [1999]).

Dysarthric Speech. Speakers who have trouble articulating, such as dysarthric[8] speakers, have problems being recognized (see for example Patel and Roy [1998]; Sawhney and Wheeler [1999]; Li and Russell [2002]; Green et al. [2003]).

Children's Speech. Children, whose vocal tract length is quite different to that of adults, are poorly recognized by systems trained on adult speech (see for example Pontamianos et al. [1997]; Das et al. [1998]; Giuliani and Gerosa [2003]; Pontamianos and Narayanan [2003]; Hagen et al. [2003]; Yildirim and Narayanan [2003]).

Older Peoples' Speech. As Wilpon and Jacobsen [1996] indicate, recognition problems also arise with the voices of the elderly.

Noise in the Signal. Ambient noise or unexpected side noise is well-known to cause recognition problems (see, for example, pages 305–309 in Rabiner and Juang [1993]).

Essentially, anything that presents sounds to the computer that differ from the models it associates with particular phonemes can cause speech recognition errors.

2.4.3 Prosodic Work

The Changes Caused by Misrecognitions

As mentioned in the introduction to this section, many researchers have suggested that users of spoken dialogue systems hyperarticulate once they encounter problems with the system recognizing their utterances. Oviatt et al. [1998] point out that hyperarticulation is a well observed phenomenon in human–human speech, and give many pointers to the research. Hyperarticulation is adopted by humans when faced with situations where they expect or experience a comprehension failure from their listener (talking to infants, speakers of different languages, and so on). Oviatt et al. report research designed to see if the variations in the way people speak can be measured. Working with a group of 20 native English speakers (half male, half female), Oviatt et al. investigate user reaction to errors introduced into the dialogue by misrecognitions. They use an environment where they can introduce such errors and find many significant acoustic–prosodic and phonological changes.

Duration. Utterance duration changes significantly from original utterance to repeat utterance when errors occur. During sections of dialogue when errors were running

[8] Dysarthric speech arises from imperfect coordination of pharynx, larynx, tongue, or face muscles.

at a low level, the increase was 16.5% (from 1544 to 1802 msec) and when high, 15% (from 1624 to 1866 msec).[9]

Speech Segment Duration Duration is 13% higher (1653 compared with 1463 msec) in high error periods when compared with low error periods.

Pause Duration. The total pause duration of multi-word utterances increases significantly from original utterance to repeat utterance when errors occur. During low error periods it rises from an average of 112 to 209 msec (85%). In high error periods it rises from 159 to 261 msec (64%). Individual pauses also showed significant elongation in repeat utterances.

Number of pauses. The average number of pauses of multi-word utterances increases significantly from original utterance to repeat utterance when errors occur. During low error periods it increases from 0.49 to 1.06 (116%) and during high error periods from 0.57 to 0.95 (67%).

Rate of Speech. The rate of speech decreases significantly from original utterance to repeat utterance when errors occur. During low periods of error it decreases from an average of 298 to 348 msec per syllable and during high periods of error from 300 to 347 msec. Overall, speaking rate decreases 16% during error clarification sub-dialogues.

Amplitude. Contrary to experience with human–human hyperarticulation, there is no significant change in amplitude of speech during error resolution.

Fundamental Frequency. Pitch maximum, minimum, range and average do not change significantly (save under limited circumstances during periods of high error).

Intonational Contour. Here Oviatt et al. look at the final intonational contour changing from rise to fall, or vice versa. Overall, likelihood of a final falling contour changes from 47% to 56% during periods of high error.

Phonological Alterations. 93% of the subjects, where Oviatt et al. had sufficient data for analysis, are observed to alter their speech phonologically during error clarification sub-dialogues at some point.[10] They all moved from conversational to hyperarticulated speech.

Disfluencies. The disfluencies rate drops significantly from 0.78 words per 100 words to 0.37 and 0.78 to 0.53 when moving from low to high error periods. In consequence, disfluencies became rarer during hyperarticulation.

[9] Oviatt et al. measure the incidence of errors and characterize parts of the dialogue with lower than average errors as low error periods and those with greater than average as high error periods.

[10] Phonological alterations occur on a change from hyperarticulated to conversational speech, or vice versa.

Phenomenon	% change
Pause interjection	+92%
Pause elongation	+75%
Disfluencies	−53%
Intonational-final fall	+19%
Speech elongation	+12%
Hyper-clear phonology	+ 9%
Pitch minimum	−2%
Pitch average	−1%

Table 2.1: Summary of changes in linguistic dimensions of hyperarticulation from conversational speech. From Oviatt et al. [1998].

Table 2.1 provides a summary of the phenomena encountered. These strong features, which have been revealed as being associated with user reaction to errors introduced by speech recognition, offer both opportunities and problems. The phenomena are reported by Oviatt et al. to cause further problems in recognition; equally they offer us a new range of data to use in identifying when errors have been introduced into dialogue as they can be used as predictive features when users respond to the system proceeding on the basis of an error.

Soltan and Waibel [1998] mostly confirm the differences between normal and hyperarticulated speech:

- fundamental frequencies increase during hyperarticulation,[11]
- vowel qualities change, and
- average phone duration is nearly 20% longer.

They went on to train a speech engine on hyperarticulated speech and reduced word error rates by 23% to 80.5% (when working with isolated words). This leads one to consider whether large hyperarticulated acoustic models can be created. Harnsberger and Goshert [2000] give some hope as they demonstrate that it is possible to elicit controlled hyperarticulated speech in laboratory conditions by re-prompting subjects to repeat a sentence *more clearly*.

[11] This was contrary to Oviatt's findings.

Classifying Utterances as Likely to be Misrecognized

Hirschberg et al. [1999] describe work using prosody to find misrecognitions in the hypotheses offered by the speech engine. Working on the TOOT Corpus (derived from a travel information system), they take the hypothesis with the highest confidence index and then use other features to see if they can distinguish errors. Looking only at recognized utterances, they mark them for semantic content (for example dates, departure and arrival cities).

(2.10) Question: What day do you want to depart?
Answer: I think Thursday would be fine.
Semantic value: from_date = 22/11/2006

(2.11) Question: Where are you going to?
Answer: London please.
Semantic value: to_city = london

(2.12) Question: Where are you going from?
Answer: I want to go from Paris.
Semantic value: from_city = paris

The semantic value of each answer is shown in each of above examples on its third line. Even if the computer's recognition contains some lexical error, if the semantic outcome is correct a *concept accuracy* (CA) of 1 is awarded. They look at 2067 user turns from 152 dialogues. There are 170 rejections (i.e. utterances not recognized based upon a low confidence index) and 491 turns with a CA < 1.

Hirschberg et al. then look to see if they can identify the misrecognitions by their prosodic features. Table 2.2 shows the results. Column Three shows the difference between the mean of each feature in recognized turns and the mean of each feature in misrecognized turns. They find the first five items to be statistically significant (at 95% confidence level, $p \leq .05$). The same features also show as significant in rejected turns. All of these features are associated with hyperarticulation in the literature as reviewed at the beginning of this section. However, they hand-tagged their corpus for utterances that sounded hyperarticulated to a human annotator. When they exclude such marked utterances, these features are still found in the utterances not characterized as hyperarticulated by human annotators. They conclude that some subtle form of hyperarticulation is still being identified by the acoustic features they extract, even though it is lost to the human ear.

Feature	T-stat	Mean Misrecognized Feature - Mean Recognized Feature	P
Pitch Max	5.40	28.87 Hz	0
Intensity Max	3.00	185.74	.005
Intensity Average	2.07	−32.43	0.5
Duration	10.42	2.34 sec	0
Prior Pause	5.22	.38 sec	0
Tempo	.45	.13 sec	.65
Pitch Avg	1.36	1.63 Hz	.18
Silence	.05	−.02%	.29

Table 2.2: Comparison of misrecognized vs. recognized turns by prosodic features across speakers [from Hirschberg et al., 1999].

Features Used	Error (Standard Error)
All	10.79%(.83)
Prosody, ASR Conf	12.71%(.75)
ASR Conf, ASR String	13.39%(.85)
ASR String	14.41%(.78)
Prosody	16.15%(.87)
ASR Conf	18.30%(.81)
Baseline (Underlying Word Error Rate)	25.88%

Table 2.3: Estimated error for predicting misrecognitions [from Hirschberg et al., 1999].

They then moved on to carry out machine learning experiments using RIPPER [Cohen, 1996]. They included a number of features which might reasonably be expected to be automatically available to spoken dialogue systems:

- the prosodic features which appeared to have predictive qualities;
- the automatic speech recognizer's confidence score; and
- the actual string recognized.

In addition, they included some features not likely to be so available, such as hyperarticulation, which was labeled by hand as noted above.

Table 2.3 shows that using all these factors predicts errors with an error rate nearly half of the baseline of 25.88%. The baseline is arrived at by simply accepting the

Features Used	Error	Standard Error
All ASR, Prompt	22.77%	.59
Prosody, All ASR, Prompt	23.66%	.80
Prosody, String	23.70%	.63
Confidence, String, LM	23.77%	.87
All features	23.91%	.85
Prosody, Confidence, LM	24.07%	.83
Prosody, Confidence, String, LM	24.19%	.94
Prosody, Confidence	24.35%	.87
Confidence, LM	25.68%	.78
Confidence	26.14%	.80
prosody	26.17%	.73
% Silence	31.30%	.93
Tempo	31.45%	.92
String	32.94%	.91
Prosody Normalized	36.31%	.79
Majority Class Baseline	39.67%	

Table 2.4: Estimated error for predicting misrecognitions [from Hirschberg et al., 2000].

recognition to be correct. What is quite surprising is the fact that the actual recognized string is predictive. They simply take the string as a bag of words and add each word as a feature. They find the presence of 'help' and '8' figure in the machine learned rules. When considered on its own, the string is a better predictor than prosody on its own.

Hirschberg et al. [2000] went on to see if this work might be generalized, and this time used the W99 Corpus.[12] Although the domain is very different, the results for the prosodic features are similar to the TOOT study. Once again they study various features using RIPPER, and the outcome is shown in Table 2.4. Again the actual string recognized is a significant predictor.

Gabsdil [2003] use two machine learning techniques, TiMBL [Daelemans and Hoste, 2002] and RIPPER [Cohen, 1996], and use the automatic speech recognizer outputs together with some ancillary information and a range of acoustic features:

- **Recognizer Confidences:** Overall confidence score, maximum, minimum, and range of individual word confidences, descriptive statistics of the individual word confidence.

[12]The W99 Corpus was derived from a system used to support registration and information access for the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'99).

- **Hypothesis Length:** Length of audio sample, number of words, syllables, and phonemes in the recognition hypothesis.
- **Tempo:** Length of audio sample divided by the number of words, phones, and syllables.
- **Recognizer Statistics:** Time needed for decoding.
- **Site Information:** At which site the speech file was recorded.
- **Pitch Statistics:** Mean and maximum F0, variance, standard deviation, and number of unvoiced frames.
- **Intensity. Statistics:** Mean and maximum RMS, variance, standard deviation, number of frames with RMS < 100.

Both methods improve classification on a baseline of confidence alone, which produces an F-score of 63.57%. At best, Gabsdil [2003] achieves 68.44% using TiMBL and 68.60% using RIPPER.

The general approach of predicting errors using acoustic features, often augmented with other information, has itself been the subject of research. Shriberg and Stolcke [2001] represents a particularly good exposition of the method of automatically discovering errors mainly using acoustic features: a corpus is hand-tagged for correct and false recognitions, and features, such as speaking rate, intensity, pitch, and durations of silence, are automatically extracted from the sound files of the associated utterances. Some machine learning technique is then employed using the tagged data and features as a training set. Hirschberg et al. [2004] reports on two studies of this nature using RIPPER [Cohen, 1996]. It is normal, in these studies, to include automatic speech recognition outputs such as confidence, but, excluding such features, they manage to achieve between 74 and 80% correct prediction that an utterance was likely to be misrecognized in two corpora. Both systems produce a correct hypothesis in around 60% of recognitions.

We have already mentioned that Hirschberg et al. [1999] show that, when they exclude utterances characterized as hyperarticulated by human listeners, they still find they can predict errors. Although they conclude that some subtle form of hyperarticulation is still being identified, on the face of things, it is likely that the human ear is the best judge of prosody. There is literature (for example [Doddington et al., 1998]) that builds upon the lore that there are some speakers (*sheep*) that speech engines find easy to recognize and some (*goats*) that they do not. Doddington et al. actually look at speaker verification, not recognition, as does the strand of literature it spawned. However, within their domain they are able to confirm that there is a speaker-related

problem, rather than a problem arising from prosodic variation from one utterance to another by the same speaker; that is, they show sheep and goats exist.

Hirschberg et al. [1999], who use prosodic features to classify utterances that will be incorrectly recognized, consider Doddington et al.'s work. They study eight speakers who produced more than 50% errors and five who produced less. They use pitch, maximum intensity, turn length and the length of silence before the person spoke as features and perform a t-test. Their results indicate that goats may be distinguished from sheep prosodically, although the simple fact that goats, as classified in their experiments, are often misrecognized did not account for all the errors they encountered.

2.4.4 Review

In this section we reviewed the literature relating to the way variations in the acoustic signal might cause and be associated with recognition errors. Firstly we saw that there can be many causes of errors arising from acoustic factors including:

- pronunciation variation,
- cold speech,
- dysarthric speech,
- children's speech,
- older peoples' speech, and
- noise in the signal.

Then we saw that there are certain principled changes in the prosodic features of speakers when recognition errors occur [Oviatt et al., 1998; Soltan and Waibel, 1998]. These are such things as increases in pause duration, slower rates of speech, increased amplitude and increased final falling contour; generally, these are attributed to hyperarticulation occurring during utterances likely to be misrecognized.

Next, we saw that there is considerable work demonstrating that one can classify utterances as likely to be misrecognized by extracting features from their acoustics. Hirschberg et al., Gabsdil and other researchers use some different features, but many of the features are common, such as:

- pitch,
- intensity,
- duration, and
- tempo.

While the performance of these classifications varied widely they achieved accuracies of between the low 60%s, up to, in one case, nearly 90%. Again the main hypothesized basis for the phenomenon is hyperarticulation. However, we also report on work by Doddington et al. [1998] which pursues the hypothesis that some people speak in a manner that recognizers find difficult to recognize all the time and others do not, so-called *goats* and *sheep*. So while it is established that one can predict errors to some extent by the use of acoustic features, it is an open question in the literature whether the errors arise from the way speakers change the phonetics of their utterances from one turn to the next in a dialogue, or if there are acoustic qualities of some speakers that generally make them more difficult to recognize. We resolve this question in Chapter 3.

2.5 Errors Arising in the Language Domain

2.5.1 Introduction

In Section 2.4 we considered work concerning misrecognitions caused by differences between the acoustic model being used by the recognizer and the sounds presenting themselves to it. Our second problem lies in the use of unexpected language by the speaker. Utterances that contain language that is not within the system's language model are often referred to as *out-of-language* utterances. All recognizers use some language model to guide the recognition. If a user employs language not covered by the model, a recognition error occurs. Unlike work in the acoustic domain, where tools other than the automatic speech recognizer are employed to extract features from the acoustic files, here the work concentrates on using speech recognizer outputs directly, often having modified the language models to allow for greater flexibility. Cuayáhuitl and Serridge [2002] provide an interesting summary of the problem. They state that there are two common approaches:

- Working with existing language models but adding generic *filler* words to cope with out-of-language utterances. Commonly, these filler words are composed of convenient sub-units of language such as phonemes.
- Working with large vocabulary recognizers in an attempt to capture all language. Often such approaches use a grammar-based recognizer focusing on the main stream of the dialogue together with a recognizer with a larger n-gram based language model to capture utterances that are off-topic or otherwise unusual.

Section 2.5.2 reviews work on using sub-word units. Section 2.5.3 reviews work done using large vocabulary recognizers in tandem with grammar based recognizers.

2.5.2 Sub-word Language Modeling

The total number of distinct words in the English language is debatable, but it is clearly very large. Owing to the largeness of the number of words multiplied by the large number of ways in which they can be associated, it is very unlikely we can create language models that can cover all the English language based upon words. However, just as the alphabet from which these words are created is limited to 26 letters, the sounds that make up the words come from quite limited sets of sub-word units. Syllables and phonemes are obvious candidates, but there are other sub-word units that will be discussed below. By using a language model that is based on language units smaller than words, we can potentially produce a speech recognizer not tied to any size of vocabulary or form of syntax.

Syllables And Phonemes

Meliani and O'Shaughnessy [1996] work with both phoneme-based and syllable-based units. They work particularly with new-word detection and keyword spotting. They experiment both on an air travel information service corpus and the Wall Street Journal. Keyword spotting consists of identifying the keywords in an utterance, such as the name of a destination city, while ignoring the rest of that utterance. Keyword spotting is not of direct interest to us, but new word detection is. Here they achieve detection rates of around 70% with models based upon phonemes and even better performance at around 80% for models based upon syllables. Filler words made up of phonemes allow any patterns of phonemes to be present; syllables constrain the patterns of sounds more closely to those actually occurring in speech. Meliani and O'Shaughnessy suggest this accounts for the increased recognition accuracy.

Boros et al. [1997] appear to use simple phoneme-based models. Their objective is to identify words of a particular semantic class such as city, region or surname, rather than the actual word itself. They work on a German corpus of train timetable enquiries. The basic idea is to build language models that are based upon word categories: for example, one for cities, one for times and so on. The probabilities of encountering out-of-vocabulary words are then established by word category and can be estimated by studying the rate of introduction of unknown words. The parameter is set at zero for groups of words which come from closed classes, but at the estimated level for groups of words in particular semantic classes, such as city, which are very large. The purpose of this work is to handle cases where, for example, a train departure time to an unknown city is requested. Once spotted, they allow the user the opportunity to spell the new city name. They only achieve a precision of 30.7% in identifying out-of-language words.

Bazzi and Glass [2000] report on work augmenting Hidden Markov Models with a phoneme based generic out-of-language word: they add an extra path to their language

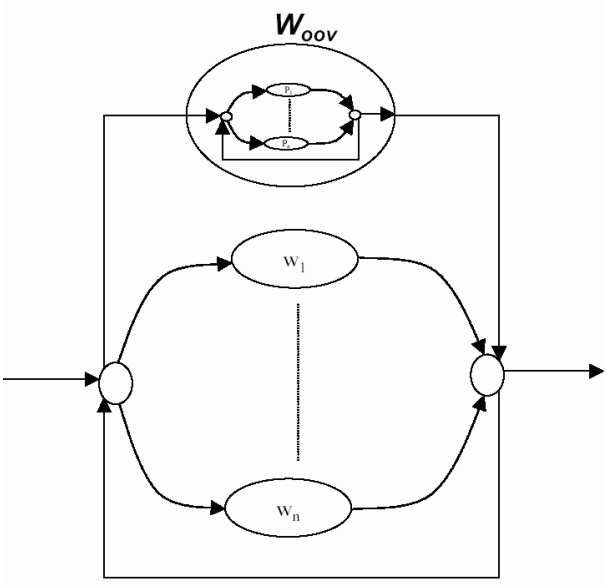

Figure 2.5: An out-of-vocabulary word placed in the recognition graph [from Bazzi and Glass, 2000].

model allowing words made up of any pattern of phonemes to be recognized. The architecture is shown in Figure 2.5. It can be seen that a path through the graph can remain within the allowed language model (W_1 to W_n) or exit into the out-of-vocabulary path W_{OOV}. They use a variable to determine the cost of entering and leaving the out-of-vocabulary path. They experiment on a corpus from the weather information domain[13] and, dependent upon the setting of the penalty, could detect between 46.8% and 54.4% of the out-of-language words. They take the work further in Hazen and Bazzi [2001], where they compare the use of confidence and the out-of-language word. The use of an out-of-language word proved superior in spotting new words; for example, at a false acceptance rate of 20%, confidence spots these words correctly 90.1% of the time and their technique achieves 93.1%.

Decadt et al.'s [2002] principal objective is to produce words from sequences of phonemes obtained when using a phoneme-based language model. They experiment with various sizes of n-gram. The best results are achieved using 5-grams of phonemes. Such a large context would be quite impossible with words owing to the sparse data problem.

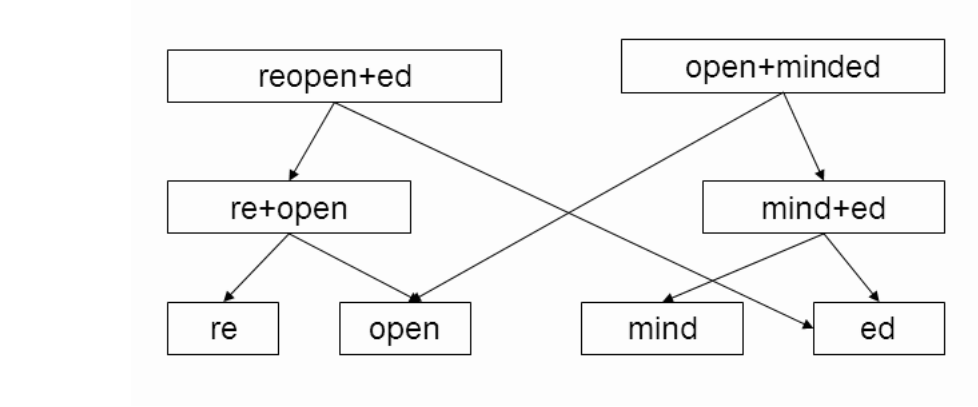

Figure 2.6: The morphs derived from a splitting tree for the words *reopened* and *openminded* [from Siivola et al., 2003].

Morphs

Siivola et al. [2003] use an algorithm to discover sub-word units called morphs across their training set, and use these units for their language modeling. The word *morph* is derived from *morphological*: the approach assumes that one can find stems, prefixes and roots, the smallest meaning bearing units of language, that through combination can account for the diversity of a vocabulary in a compact manner. Figure 2.6 shows how the two words *reopened* and *openminded* can be accounted for by their morphs *re*, *open*, *mind* and *ed*. These morphs can be used instead of phonemes or syllables as the basis for a language model.

Siivola et al. run a number of experiments on material from talking books using three language models: word based, syllable based and morph based. Error rates dropped progressively from 50.4% with words to 43.9% with syllables and 31.7% with morphs. They argue that the approach was particularly worthwhile in Finnish,[14] a language rich in morphology, and would benefit work in, for example, Turkish or Hungarian.

Comparison of Various Methods

Facco et al. [2006, page 205] report on six experiments comparing various types of rejection grammars, that is, grammars specifically designed to spot out-of-language utterances. The performance of the various types is shown in Table 2.5. They work on a corpus derived from a banking system. They do not fully define the nature of the methods, but the phone loop would appear to be a simple context independent model of any pattern of phonemes. It produces the worst word error rate of 27.4%. They

[13]The corpus was derived from JUPITER, a spoken dialogue system developed by MIT [MIT, 2006].
[14]Finnish is the language in which Siivola et al. [2003]'s experiments were carried out.

Rejection Grammar	Sentence Error Rate	Word Error Rate
phone loop	27.4%	36.7%
tuned phone loop	20.6%	26.5%
garbage lexicon	13.7%	17.4%
garbage unigram	12.9%	16.2%
bigrams	9.1%	12.9%
trigrams	8.0%	11.1%

Table 2.5: Performance of different rejection grammars [from Facco et al., 2006].

then optimize the probability of encountering out-of-language utterances by tuning a parameter associated with entering the phone loop, and improve performance to 26.5%. They then built a lexicon of all the words in their development sets not belonging to in-language utterances. They used this in the progressively more structured ways shown in the table: in the first two cases (phone loop and tuned phone loop) essentially as a bag-of-words, then as a garbage lexicon and finally as garbage unigrams, bigrams and trigrams. The introduction of the extra structure continues to improve recognition accuracy. Essentially the phone loop and tuned phone loop are two types of filler model; the garbage lexicon and garbage unigram, bigram and trigram are examples of large language models.

2.5.3 Large Vocabulary Recognizers

Many spoken dialogue systems are now built using grammar-based recognizers. Designers can build a language model that concentrates on the immediate focus of the dialogue quickly and simply when using such recognizers. It is relatively easy to add an n-gram recognizer designed to capture a wider range of speech. Comparison of the two recognitions, one from the grammar-based recognizer and one from the n-gram based recognizer, can then be used to try to identify out-of-language utterances. There is some work published exploring what can be learned from the differences between two recognitions, one grammar-based and one n-gram based, of the same utterance. The next two papers report on improving the help available to users by using this technique.

Gorrell et al. [2002] report on a system that controls the functions in a house. This is a system that allows the user, using speech, to turn the lights on and off in various rooms, or the computer, VCR or television and so on. A grammar-based system controls the basic application, but a large corpus of failed utterances is collected. Features such as the words used, their number, their confidence scores and so on are extracted and used to train a classifier that produces 12 groups of such utterances that are amenable

to a simple response by the system to the user such as:

(2.13) *I didn't quite get that.*

(2.14) *Long commands are difficult to understand.*

(2.15) *Perhaps try giving separate commands for each device.*

A large vocabulary recognizer is run in tandem with the main grammar-based recognizer and the same features extracted from the proposed hypothesis and appropriate help offered to the user.

Subsequently, Gorrell [2003, 2004, Chapter 5,] reports on further work done on spotting out-of-vocabulary words using the same corpus. In this work she ran a grammar-based recognizer and an n-gram based recognizer in tandem. She accepts any utterances recognized by the grammar-based recognizer with a confidence of over 45% as correct, but otherwise looks to see if words appear in the n-gram recognition that are outside the grammar's vocabulary. If they do, and the word confidence is over a particular threshold (she tries various thresholds between 30% and 60%), she accepts she has found an out-of-language utterance. She concludes that she can spot about 50% of out-of-vocabulary words.

Hockey et al. [2002] use a similar approach in a rather different domain. Here they endeavored to improve dialogue with a robot. The problem of providing help to the user is augmented by the use of two recognizers: one grammar-based and one n-gram-based. The form of help here is to provide the user with advice as to what has actually gone wrong (for example, an unknown word or unrecognized syntax) together with an example of an in-grammar utterance based upon an analysis of the n-gram-based recognition.

Finally, López-Cózar and Callejas [2006] use the technique of creating a second language model based upon word-class bi-grams derived from language expected to be used in response to other parts of a dialogue than the current focus. This model was used to determine if the speaker was actually providing the answer to some other question the system would ask in the dialogue and the probabilities associated with this model were used to re-score the proposed hypotheses offered by a more restricted language model that only covered the language expected in response to the prompt at that point in the dialogue. The approach allowed the system to understand eight out of ten off-topic utterances by users with a loss of only 2.6% in word accuracy for on-topic utterances.

2.5.4 Review

In this section we reviewed the way in which the use of unexpected language by speakers can be identified and handled. The work concerning identifying out-of-language

utterances is based upon two principal approaches:

- modeling language using some limited set of elements such as phonemes, or
- using a second recognizer trained on a more general language model.

In both cases more general language models are created than normally used for recognition and various features of these are used as clues to identify the presence of out-of-language utterances. For example, in the former case, a generic word might be created using phonemes and if it, rather than word(s) otherwise contained in the original grammar model, is returned as a recognition, it is assumed an out-of-language utterance has been encountered. In the second case a comparison of a second recognition might reveal words not in an original language model and this might to indicate the presence of an out-of-language utterance.

In Section 2.5.2 we looked at using syllables and phonemes as the appropriate sub-word unit [Meliani and O'Shaughnessy, 1996; Boros et al., 1997; Bazzi and Glass, 2000; Decadt et al., 2002b]. Then we look at using morphs for the same purpose [Siivola et al., 2003]. Finally, in that section, we look at a comparison of the accuracy of the various methods [Facco et al., 2006].

In Section 2.5.3 we looked at work using a second large vocabulary recognizer [Gorrell et al., 2002; Gorrell, 2003, 2004; Hockey et al., 2002].

As with acoustic errors, we saw that there are techniques that can identify the likely presence of out-of-language utterances with some degree of predictive accuracy. Although predictive accuracy varied widely, using an out-of-grammar path in a grammar model achieved around the 80% level.

2.6 Conclusions

In this review we first looked quite widely at the problems that recognition errors cause in spoken dialogue systems. In Section 2.2 we placed the focus of this thesis, early stage detection, in context, briefly looking at error prevention, early and late detection and recovery.

Our review of the literature on errors and how to handle them showed that this is a very large field. Problems still exist at all stages of the process: recognition hypotheses are all too often incorrect, and systems have no reliable way of knowing this; once errors enter the dialogue, systems and users often stumble about in repair sub-dialogues sometimes only compounding the problems; there are many encouraging strands of research as to how one might go about repairing a dialogue when it has got into trouble, but all are inconclusive and require work to be brought to fruition in practical applications. The field is obviously ripe for considerable further research.

We decided to concentrate on one single aspect of these problems: early stage identification of speech recognition errors. The current state of the art in this respect is the use of confidence, so in Section 2.3 we looked at how it is used, calculated and performs as a classifier.

Speech recognizers, at least conceptually, tackle the problem of recognition by dividing it into two distinct areas: the aspects of how words sound and the aspects of what words are used in utterances. The literature relating to early stage detection of speech recognition errors can also be reviewed in this fashion:

- In Section 2.4 we looked at work in the acoustic domain: first, at the changes that appear to happen in speech when errors occur; then at the extensive work that has been done using acoustic features to predict error-prone utterances.

- In Section 2.5 we looked work in the language domain: first at modeling generic words using sub-word units such as phonemes to spot out-of-language words; then at using a second recognition based upon a large language model for the same purpose.

The literature encourages us to believe that the problem of early stage detection of speech recognition can be substantially solved. In the acoustic domain, there is work by Hirschberg, Scriberg, Stolcke, Litman and others indicating that one can predict if an utterance will be misrecognized from acoustic data (length, pitch, intensity, etc). In the language domain, there is work by Bazzi, Glass, Hazen, and Kwok indicating that one can model a meta-word to capture all language and place it in the recognition path and predict if out-of-grammar utterances are being encountered. In addition, there is work by Gorrel and Hockey on using secondary recognition with large vocabulary recognizers to catch out-of-language utterances and improve overall system performance.

We know that incorrect recognitions are offered by recognizers when faced with different sounds or words to those in the models they employ, so this work encourages us to believe that it might be possible to study methods to identify when errors occur in either domain, if possible improve on the existing position, and then combine the results into a unified classifier that can predict if an utterance would be misrecognized for either reason. The very considerable work that has been done on acoustic errors points the way to a statistically-based classification task. In addition, it may be that further insights into the cause of acoustic errors can be discovered. We have already mentioned that there are two hypotheses concerning prosody as the source of errors: one that favors the way people change prosody while speaking and the other that favors the idea that some people produce speech all the time that is simply difficult to recognize. In the language domain the work on spotting out-of-language utterances remains incomplete with considerable research required into the various methods involved.

As can be seen from this review, most research has favored either working on acoustic errors or language errors, but research has not tried to bring together both aspects into a unified model designed to significantly raise the prediction accuracy of confidence.

In order to do so we have to bring together and develop the existing work in the acoustic and language domains and then move on to create a unified model. This is pursued in this thesis:

- In Chapter 3 we investigate the causes of acoustic errors and our ability to automatically spot them.
- In Chapter 4 we investigate the possibilities of discovering out-of-language utterances automatically.
- In Chapter 5 we apply a range of machine learning techniques covering the data from both domains to produce a unified classifier and evaluate it on real-life data from a fielded system.

Chapter 3

Errors in the Acoustic Domain

3.1 Introduction

In this chapter we report on our work in the acoustic domain. Of course, it is central to the task of classification to demonstrate that, in this domain, we can determine if an utterance is likely to be misrecognized purely on the basis of its intrinsic acoustic features. To do this we require a particular type of corpus; one free of language errors and side noise. TIDIGITS provides us with studio-quality recording and, because all the utterances are sequences of spoken digits, the ability to write a grammar that covers all possible utterances.

The idea of using acoustic properties to detect the likelihood of error is not in itself new; Hirschberg et al. [2004], Shriberg and Stolcke [2001], Litman et al. [2001] and others have explored this area, based on the hypothesis that specific properties of particular utterances, in particular hyperarticulation, are what give rise to recognition errors. However, this begs what is still an open question in the literature: *Is it the way a person changes the way s/he says one utterance from another that causes recognition problems or the way a person speaks generally?* The second possibility is referred to generally as *the sheep and goats hypothesis*. Doddington et al. [1998] show that this hypothesis holds in the context of speaker verification, spawning a string of literature on that topic.

Finally, working in the acoustic domain offers us the opportunity to better understand which acoustic features cause recognition problems. We look at the effect of novel features, such as those associated with speech pathology, which have not been included in acoustic feature sets before.

The working assumption in this thesis is that we can usefully distinguish two primary kinds of errors in recognition: errors that arise from a mismatch between the recognizer's acoustic model and the characteristics of the speaker's voice, and errors that arise from a mismatch between the recognizer's language model and the words uttered by the speaker. This chapter concentrates exclusively on the former area, the acoustic domain,

leaving the language domain to Chapter 4.

First, in Section 3.2 we do some preparatory work looking at modeling and classification techniques. We concentrate particularly on logistic regression, the technique we employed. Then in Section 3.3 we turn to a series of hypotheses we test in our experiments. The hypothesis are designed to allow us first to show we can predict which utterances will be misrecognized and secondly to establish that the sheep and goats hypothesis holds true for speech recognition. The section goes on to explain the materials used and the methods. The experiments were first carried out on parts of the TIDIGITS Corpus and these are extensively reported on in Section 3.4. Section 3.5 reports on the similar set of experiments on the CU Corpus, undertaken to verify the results using a very different corpus.

In the second part of the chapter at Section 3.6 we turn to investigate why it is unlikely that prosody can play much part in the recognition process when using current state of the art recognizers, and report on our findings on the causes of acoustic errors. Finally, in Section 3.7 we review the outcomes of our work in the acoustic domain.

3.2 Modeling and Classification

Statistical studies are often used in scientific research. Machine learning is an area that has grown in popularity with the easy availability of computing power, and we make extensive use of such techniques, which are reported on later in this thesis (Chapter 5). Initially we sought to investigate the relationships between a wide range of acoustic features (26 in number) and errors in recognition. This is a classical statistical problem and there are a number of techniques that one might use to investigate the relationship between predictive features and outcomes. Principal components analysis (see for example Jolliffe [2002] for a comprehensive introduction) is one such technique for identifying the important factors in determining outcomes, but we are very interested in being able to predict classification in addition to investigating the relationships of the factors. We chose to use logistic regression, a well established technique infrequently seen in the computer science literature, but well reported in the statistical literature (see, for example, such leading works as Ripley [1996]). Logistic regression is a technique used extensively in epidemiological studies and provides a method for studying the relationships between predictive features and some binary (or at least categorical) outcome. It is supported by most statistical software packages. We use this technique extensively in our work and review it in this section.

3.2.1 Logistic Regression

In epidemiological studies the problem is to discover the relationships between a number of independent variables and some disease outcome. In our case it is to discover the

relationships between a number of features present in the sound (F_1, F_2 up to F_k) and recognition without any errors (C). This probability can be represented by $P(C = 1|F_1, F_2...F_k)$, more briefly referred to here as P(**F**). This can be modeled as:

$$P(\mathbf{F}) = \frac{1}{1 + e^{-(\alpha+\Sigma\beta_i F_i)}}$$

where the terms α and β represent unknown parameters that must be estimated on the basis of our data. When using linear regression, these parameters are estimated using the least squares method. Although different techniques have been used in the past, the technique currently favored to estimate these parameters for logistic regression is called *maximum likelihood estimation* [Aldrich, 1997].

Maximum likelihood estimation strives to find the parameter value(s) that make the observed data most likely. Normally, we endeavor to make predictions based on a set of observations by using probabilities, the probability of certain outcomes occurring or not occurring. In the case of data analysis, we have already observed all the data, and once observed, it is fixed. This allows the use of a numerical optimization method to estimate α and β. This method of estimation can be used for models of great complexity, and software packages such as R (see Section 1.3.1) come equipped with algorithms that carry out the maximum likelihood estimation for us and produce our unknown parameters, α and β.

Logistic regression is very robust and does not carry with it the assumptions of a normal distribution that are employed in linear regression. Logistic regression allows us to explore the effect of each feature on the outcome, and to predict the outcome. Once the logistic function has been calculated, we can estimate the sensitivity of the model to each feature by entering relatively high and low figures for each feature in turn.

Generalized linear models (of which logistic models are one type) are now being used more generally outside their initial fields of application. Mazerolle [2004] looks at habitat loss and reduction in global amphibian populations using such techniques and (in his Appendix I) reviews Akaike's Information Criterion (AIC) [Akaike, 1973], which is used to handle model selection and inference.

The mathematical definition of AIC is shown at Equation 3.1:

$$AIC = -2lnML + 2k \tag{3.1}$$

where ML denotes maximum likelihood, lnML is the value of the maximized log likelihood function for a model fitted to a given data set, and k is the number of independently adjusted parameters within the model.

Where one commences a study with a large number of potentially predictive factors, one not only faces the problem of estimating the effect (magnitude) of a given variable on

the response variable, but also of assessing whether the effect is sufficiently important to include the parameter in the model in order to make predictions. An approach developed in the early 1970s rests on AIC and its associated measures. It is known as the information-theoretic approach, as it arose from information theory (see Kullback and Leibler [1951], Cover and Thomas [1991] and Burnham and Anderson [2001]).

Parsimony is a particular issue in model building; a compromise has to be made between model bias and variance, where these two factors are defined as:

Bias: The difference between the estimated value and true unknown value of a parameter.

Variance: The precision of the estimates.

As Burnham and Anderson [2002] point out, a model with too many variables will have low precision whereas a model with too few variables will be biased. Models only approximate reality, so in their construction we are trying to minimize the loss of information. Kullback and Leibler [1951] developed a measure for this, the Kullback-Leibler Information Criterion, to represent the information lost when approximating reality. Akaike [1973] proposed using the Kullback-Leibler Information Criterion for model selection. He established a relationship between maximum likelihood and the Kullback-Leibler Information Criterion. The AIC metric can be used to rank models from best (lowest) to worst (highest) and packages such as R have functions that work through the parameters in a model and discard them in a stepwise fashion so long as the AIC is improved (reduced), leaving one with a 'best' model. We used AIC to reduce our logistic models to their more significant sets of features.

3.2.2 Evaluation

Logistic regression affords us the opportunity to make predictions using sets of values for the features. When predicting the classification of any phenomenon, it is important to consider the role that chance may play in the outcome. The problem with simple direct comparison of the prediction with the outcome is clearly shown in a case where 95% of the utterances are correctly recognized. A classifier that simply predicts all the utterances are correctly recognized will have a success rate of 95% simply by chance. The potential for a 'misleading' level of success increases the more the underlying data is skewed.

To cope with this problem, it is common in many fields to use the metrics of precision and recall, and their combination as some form of mean in an F-score to measure the success of classifications. This metric has also been used in the field of error prediction (see, for example, Gabsdil [2003]), but we chose to use prediction accuracy (that is the proportion of things that were correctly identified) together with the Kappa statistic.

These are the preferred statistics used when working with logistic regression, as we mainly did for investigatory work, and, therefore, allowed us to look to the published literature when interpreting results (see, for example McNeill [1998]). Of course Kappa cannot be used in cases where there are not clear negative case counts and this is often so in question-answer tasks or document retrieval. Consider the case of searching for documents on the Internet: negative cases correspond to all non-relevant Internet documents. Their number is very large, poorly defined, and constantly changing; Kappa would not be a useful metric [Hripcsak and Rothschild, 2005].

Many researchers who come across problems involving annotation will have seen the Kappa statistic used to compare the similarity of one annotator's mark-up of a corpus with another, as against the similarities which would arise by chance [Carletta, 1996]. However, it may be equally as usefully employed to compare a prediction against a chance outcome and is commonly used in epidemiological studies and other statistical studies for this purpose. Indeed the R package e1071 comes with it as a standard function (classAgreement) for just this purpose [Dimitriadou et al., 2004].

The Kappa coefficient (K) measures pairwise agreement, across the predicted outcome and the underlying distribution, correcting for the expected chance agreement. The formula is:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) is the proportion of times that the prediction agrees with the outcome and P(E) is the proportion of times that we would expect them to agree by chance. When there is no agreement (other than that expected by chance) the Kappa is 0. When there is total agreement the Kappa is 1. In the field of dialogue mark-up, researchers look for a high Kappa (even over 80%) to characterize agreement between different annotators [Krippendorf, 1980], but when simply used to indicate if a classification is reliable, for example in epidemiological research, much lower levels (30 and 40%) appear to be accepted as indicating an outcome removed from chance (see for example Tezuka et al. [1992]).

We accept that the use of Kappa, and in particular using it as an indication of reliability, at lower levels than are usual in annotation tasks may lead some readers who work principally in that area to doubt some of the conclusions we come to as a result of our experiments. However, we believe that we are justified in doing so as clearly there is a considerable difference between an exercise where one is trying to see if two humans are consistent in deciding if something falls into a class, for example, if a word is a referent for an anaphora, and if medical symptoms are associated with, for example, death. Subject to human error, the annotations against which we are working, are as objective as an event such as death and therefore one of our classifications is objective

and levels of Kappa do not have to take into account half of the problem of human error that are present in annotation exercises.

3.3 Hypotheses, Methods and Materials

3.3.1 Introduction

In this section we look at the hypotheses, methods and materials we used in our work in the acoustic domain. In Section 3.3.2 we explain the four hypotheses our experiments test. First we want to establish the extent to which we can classify error prone utterances from their acoustic features. This is captured in two hypotheses: Hypothesis 1 concerns consistency and Hypothesis 2 concerns prediction. Hypotheses 3 and 4 pursue the undecided question: do acoustic errors arise from changes in individual speaker prosody (the hyperarticulation position) or from problematic speakers (the sheep and goats position)?

In Section 3.3.3 we review the materials of our experiments. We look at the two corpora TIDIGITS and CU. In Section 3.3.4 we review the methodology used involving running recognitions, classifying the results as correct or not, extracting acoustic features, model building and evaluation.

In Section 3.3.5 we explain the acoustic features we extract.

3.3.2 Hypotheses Tested

We tested four hypotheses:

Hypothesis 1: The features that predict errors are consistent throughout the corpus.

To detect errors at an early stage we are most interested in being able to predict if new utterances can be classified as correctly recognized. The data we normally have to do this with is a new utterance and a training set of previously recognized utterances. Therefore, it is critical that the features that predict errors hold over different speakers. We can set up an experiment (*the Consistency Experiment*) to test on every tenth utterance and train on the balance, and by moving one utterance ahead each time we can run ten tests. The results will test the first hypothesis.

Hypothesis 2: Models trained on one group of speakers will accurately classify the utterances of different speakers.

In the previous test we will often be training and testing on utterances of the same speaker as each speaker contributed about 70 utterances to the corpus. In the real world we will rarely have encountered the speaker of the test utterance before. We can set up an experiment (*the Prediction Experiment*) where we partition the data so that

we can train and then test on utterances from different speakers. The results will test the second hypothesis.

Hypothesis 3: Errors are not uniformly distributed over speakers.

If errors arise as a consequence of acoustic properties of the utterances of particular speakers they will occur more often than can be accounted for by a chance distribution in certain speakers. This will be particularly so in a corpus such as TIDIGITS where the speakers are not engaging in a dialogue. This should hold even though in this case all the speakers are native American-English speakers chosen to be dialectically balanced. The third experiment (*the Error-Prone Speaker Experiment*) is designed to test this.

Hypothesis 4: The models, once trained, will classify which speakers present with recognition problems.

Finally, if our model is good enough to predict which utterances have been misrecognized, will it also classify which speakers presented with recognition problems? Our fourth experiment (*the Goat Experiment*) is designed to test this.

The four experiments carried out to test these hypotheses are reported on in Section 3.4 when carried out on the data sets from the TIDIGITS Corpus, and in Section 3.5 when repeated on the data sets from the CU Corpus.

3.3.3 Materials

The experiments were carried out on parts of the two corpora, TIDIGITS and CU. These corpora were introduced in Section 1.3.2. The significant difference between these corpora is that TIDIGITS is made up of studio quality recordings of spoken sequences of digits while CU represents over-the-phone-quality recordings of spontaneous speech arising during travel information dialogues.

It transpired that we could not simply rely upon the annotations that came with TIDIGITS: it turned out that a number of the sound files were subject to early truncation with the effect that some annotations appeared to be for utterances longer than the sound file contained. This made any recognition incorrect when compared with the annotation. We subjected the testing section of the corpus to recognition using Sphinx4 and then checked each case which was misrecognized.

We excluded some 574 (4.8%) of the utterances on the grounds of early truncation. Our first experiment was to test the consistency of the predictive features across the entire corpus, so we used the balance of the utterances (some 12,000) for the Consistency Experiment. We then wanted to test if one could derive predictive features from one group of speakers and predict if utterances from other speakers would be misrecognized. We made up two sets each of 4000 utterances, each coming half from men and half from women, and used them for the Prediction Experiment. The balance of the experiments were run on men's speech (just over 4250 utterances). Generally, at any given sampling

rate a man's speech contains more formants than a woman's speech and the loss of data that arises from the loss of formants may be responsible for the poorer recognition rates we experienced with the female speakers in the TIDIGITS Corpus when compared with the male speakers. Choosing male speakers was designed to increase recognition accuracy.[1] Recognitions were run against both the TIDIGITS and WSJ acoustic models to see how different acoustic models. affected the results.

The CU Corpus contrasts with TIDIGITS, being richer in the variety of utterances, language employed and syntax. However, many of the dialogues in the corpus only run to one or two turns. This arises from the fact that users of spoken dialogue systems often hang up early in the process when they face difficulties or don't know what to say. We decided to restrict the dialogues used to those with over ten utterances per speaker to obtain more data per speaker. Ultimately our data set ended up with a little over 16,000 utterances across some 418 dialogues. Of course, the removal of the shorter dialogues can give rise to the criticism that we might have biased the subset, cutting out all problematic dialogues. However, in this case, the utterance error rate actually increased marginally from 36.65% over the entire original corpus to 37.89% in the subset.

Both our corpora came with annotations, and the CU Corpus came with transcripts of the speech engine's recognitions. TIDIGITS afforded us the opportunity to create a language model that covered all utterances so only acoustic errors would be present. Furthermore, as this corpus is studio quality sound, it is free of side noise. The CU Corpus allowed us to see if the results of the experiments carried out on studio quality sequences of spoken digits would still hold when they were run on utterances from a spoken dialogue system.

3.3.4 Method

Our method comprises a number of steps designed to allow us to bring together the required data to build a logistic model and then evaluate how well that model performs:

Recognition. In the case of the TIDIGITS Corpus we produced recognitions using Sphinx4; the CU Corpus came with such information generated by the University of Colorado's SONIC recognizer [Pellom, 2001].

Classification. We then labeled the recognition as correct or containing an error by a strict string comparison with the annotations that came with both corpora.[2]

[1] Women generally have higher pitch speech. Any given sampling rate will place a ceiling on the frequency captured in a sound file [Nyquist, 1928]. This result in the loss of more formants in women's speech generally than in men's.

[2] Errors are often subcharacterized as being insertion, deletion or substitution errors. However, we are simply trying to ascertain if an utterance is misheard. For our purpose, it is sufficient to simply

/TIDIGITS/TEST/WOMAN/TB/175A.RAW one seven five
/TIDIGITS/TEST/WOMAN/TB/1844A.RAW one eight four four
/TIDIGITS/TEST/WOMAN/TB/1973OA.RAW one nine seven three oh
/TIDIGITS/TEST/WOMAN/TB/19A.RAW one nine
/TIDIGITS/TEST/WOMAN/TB/1A.RAW one
/TIDIGITS/TEST/WOMAN/TB/1B.RAW one

Figure 3.1: Sphinx4 batch file.

Extraction of Acoustic Features. We then extracted a set of acoustic features (see Section 3.3.5) from the sound files. This gave us the values for the features associated with each utterance.

Model Building. Next, we trained and simplified logistic models on some segment of the data and then used these models to classify the utterances in some other part of the corpus as correctly recognized or not.

Evaluation. Finally, we evaluated the outcome by comparing our predictions with the actual results.

The experiments commenced with large sets of sound files that came with the annotations. Figure 3.1 (page 58) shows a few lines from a batch file, where the first part of each line is the location of the sound file and the second part an annotation of the utterance contained in the sound file.

Figure 3.2 shows the output from Sphinx4 for an example utterance when run in batch mode. The first line (REF) contains the human annotation of the utterance and the second line (HYP) the recognition from Sphinx4. In this case the utterance was correctly recognized, resulting in the statistics shown in lines 3–5.

The same batch files were used, without the annotations, to run a script over the associated sound files and extract the range of features described in Section 3.3.5. We used Praat to extract the features and capture them in text files; Figure 3.3 shows a typical case.

We used logistic regression as the main method for analysis and classification. This method allowed us to discover which of the 26 features[3] were important in contributing to the results, and the sensitivity of the outcome to each of these individual features. In

characterize a recognition hypothesis as being in error if it does not exactly match the words in the speaker's utterance. Of course, our simple binary method of classifying recognitions as correct or incorrect combines many different types and levels of errors within the incorrect group. However, all of them do constitute mishearings.

[3]Table 3.1 (page 58) only shows 25 features. At the end of Section 3.3.5 we explain how we derived the 26th feature, speaking rate, from two of these features.

REF: one two nine eight
HYP: one two nine eight
Accuracy: 100.000% Errors: 0 (Sub: 0 Ins: 0 Del: 0)
Words: 4 Matches: 4 WER: 0.000%
Sentences: 1 Matches: 1 SentenceAcc: 100.000%
This Time Audio: 1.62s Proc: 2.56s Speed: 1.58 X real time
Total Time Audio: 1.62s Proc: 2.56s Speed: 1.58 X real time
Mem Total: 126.62 Mb Free: 111.00 Mb
Used: This: 15.62 Mb Avg: 15.62 Mb Max: 15.62 Mb

Figure 3.2: Examples of a recognition report from Sphinx4.

addition, it built classification functions which, once trained on part of a corpus, could be used to classify the outcomes on some other part of the corpus.

The standard procedure for logistic regression is to create a logistic model and then use some form of feature reduction to identify the more significant predictive features. We used R to create a logistic model. We then used the MASS module for R (using Akaike's Information Criterion [Akaike, 1973]) to reduce the size of the model and better fit it (see Section 3.2.1). We then used our logistic model as a classifier for utterances in a test set.

The experiments were carried out on both the TIDIGITS Corpus and the CU Corpus.

3.3.5 Acoustic Features

As the literature reveals, acoustic features that might be associated with prosody are often used in predicting errors in recognitions (see for example Hirschberg et al. [1999, 2004], Litman et al. [2001]). Some of the literature concentrates almost exclusively on hyperarticulation as a cause of errors (see for example Stifelman [1993]; Oviatt et al. [1998]; Levow [1998, 1999]). This thesis follows Huang et al. [2001, pages 739–740] in defining prosody as that aspect of spoken communication that adds to or changes the meaning of the simple lexical content of an utterance. We suspected that features that would not normally be thought of as prosodic, such as speech pathology, play a role in causing errors. We use the term *acoustic features* to distinguish this larger set of features from those contained in a set which is simply prosodic. Of course, these acoustic features include the more restricted set of prosodic features.

While pitch is a significant prosodic feature, Shin and Kochanski [2002] point out that it has been known since the 1950s that duration and intensity are also significant

prosodic features (see Fry [1955, 1958]; Bolinger [1958]; Lieberman [1960]; Hadding-Koch [1961]). In addition, more recent literature provides support for spectral tilt and jaw movement as significant generators of prosodic features (see Maekawa [1998]; Kehoe et al. [1995]; Sluijter and van Heuven [1996]; Pollock et al. [1990]; Sluijter et al. [1997]; Turk and Sawusch [1996]; Ericsson and Delaney [1999]).

The *ugly duckling theorem* [Duda et al., 2001, Section 9.2.2] states that there is no problem or purpose independent method for the selection of features that may be used to define similarity among objects for classification. The most useful set of features must be developed for the task in hand. We are interested in looking at a wider range of features than other researchers have employed. In particular, we wish to include speech pathology features, such as the creakiness in a voice, as one area that seems to have been ignored in previous research into the acoustic sources of recognition errors. The features shown in Table 3.1 are extracted from sound files for use in our experiments. This long list of features is organized in four groups:

Intensity. Intensity not only reflects the relative loudness of a speaker, but also any loss of energy due to the transmission channel, for example a telephone line.

Pitch/Formants. Pitch, F1, F2, F3, F4 and F5 all relate to the vocal equipment of the speaker.

Timing. Length, speaking rate and time to commence speaking are all temporal qualities of the utterance.

Speech Pathology. Jitter, harmonicity, voice breaks, unvoiced frames, and shimmer all measure speech pathology indications such as creakiness in the voice, background noise in the voice, and the like.

The task of calculating speaking rate is normally accomplished by establishing the number of syllables in an utterance, and then dividing the number by the length of the utterance in seconds to produce the number of syllables per second. Most researchers achieve this by estimating the syllables from the best recognition offered by the recognizer. However, it is desirable to be able to establish the speaking rate without having to recognize the utterance. There is a burst of intensity associated with the vowel that comprises part of each syllable; counting these local maxima gives an estimate of the number of syllables. We validated this measure against a count of the syllables from the recognition hypotheses. They compared well with the estimates from the recognitions. There were some 5249 syllables actually present in the data and we spotted 4656 syllables using this technique. A scatter plot and line fitted using linear regression are shown in Figure 3.4. The relationship between the two variables is significant since the p-value is less than 0.0001 (p-value: $< 2.2e\text{-}16$). The regression line explains 82.9% of the variance of the data. This algorithm therefore allows the extraction of speaking

length => 1.651
meanPitch => 141.41
minimumPitch => 110.38
maximumPitch => 155.39
meanF1 => 809.690
meanF2 => 1777.199
meanF3 => 2830.182
meanF4 => 3898.843
meanF5 => 4313.072
ratioF2ToF1 => 2.195
ratioF3ToF1 => 3.495
jitter => 0.014
shimmer => 0.071
percentUnvoicedFrames => 0.488
numberOfVoiceBreaks => 1.000
percentOfVoiceBreaks => 0.076
meanIntensity => 47.026
minimumIntensity => 9.411
maximumIntensity => 69.672
ratioIntensity => 1.482
noSyllsIntensity => 4.000
meanHarmonicity => 15.976
minimumHarmonicity => -226.589
maximumHarmonicity => 39.212
startSpeech => 0.616

Figure 3.3: The output from Praat for a single utterance.

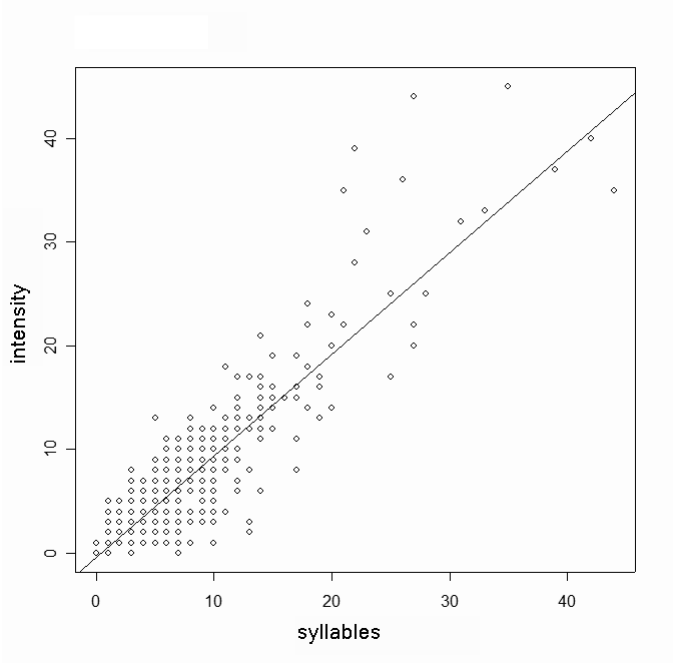

Figure 3.4: Graph confirming the correlation of estimates of numbers of syllables calculated directly from sound files with those from recognition hypotheses.

rates directly from sound files without having to go through the recognition process.[4]

3.4 The TIDIGITS Experiments

3.4.1 Introduction

In this section we describe the four experiments we carried out to test the hypotheses stated in the previous section. In order to explore the sheep and goats hypothesis we wanted a corpus that was free of side-noise and other non-speech related problems. The TIDIGITS Corpus offered the unusual opportunity to experiment on a read series of digits, recorded at studio quality and without noise in the signal. We ensured that there were no language model errors during recognition by writing a grammar that covered all the utterances. If we found errors (and we did), they could be expected to arise solely from the fact that some quality of the speaker's utterance was not represented in the acoustic models. The features revealed as being predictive might give us clues as to why these differences arose. In addition, if the sheep and goats hypothesis were true, some speakers (whose acoustic features fell outside the training set) should be misrecognized more than others, more than can be accounted for by chance. Sphinx4 was configured using the TIDIGITS build files. A jsgf grammar was employed. Triphone models were

[4]The technique still worked for short utterances but was less accurate. When we excluded all utterances with over 10 syllables R^2 fell to 38%.

Feature	Description
INTENSITY	
maximumIntensity	The maximum level of intensity achieved during the utterance in decibels.
meanIntensity	The mean level of intensity achieved during the utterance in decibels.
minimumIntensity	The minimum level of intensity achieved during the utterance in decibels.
ratioIntensity	The ratio of maximum intensity to minimum intensity.
PITCH/FORMANTS	
maximumPitch	The maximum pitch (F0) during the utterance in Hertz.
meanF1	The mean of the first formant in Hertz.
meanF2	The mean of the second formant in Hertz.
meanF3	The mean of the third formant in Hertz.
meanF4	The mean of the fourth formant in Hertz.
meanF5	The mean of the fifth formant in Hertz.
meanPitch	The mean pitch (F0) during the utterance in Hertz.
minimumPitch	The minimum pitch (F0) during the utterance in Hertz.
ratioF2toF1	The ratio of the second formant to the first formant.
ratioF3ToF1	The ratio of the third formant to the first formant.
TIMING	
length	The length of the utterance in milliseconds.
noSyllsIntensity	The number of syllables in the utterance estimated by counting local maxima in intensity.
speakingRate	The number of syllables per second estimated by counting local maxima in intensity and dividing by the length of the utterance.
startSpeech	The time in milliseconds until speech commenced.
SPEECH PATHOLOGY	
jitter	A measure of periodicity disturbances in the acoustic signal arising from variations of the fundamental frequency.
maximumHarmonicity	The mean harmonics-to-noise ratio in the signal.
meanHarmonicity	The mean harmonics-to-noise ratio in the signal.
minimumHarmonicity	The mean harmonics-to-noise ratio in the signal.
numberOfVoiceBreaks	The number of voice breaks in the voiced part of the utterance.
percentUnvoicedFrames	A measure of the number of frames during speech when voicing is lost.
percentOfVoiceBreaks	The percentage of the voice breaks (measured in time) as a proportion of the spoken part of the utterance.
shimmer	A measure of periodicity disturbances in the acoustic signal arising from variations of the cycle-to-cycle peak intensity.

Table 3.1: The features extracted from sound files.

used with three states. The acoustic model feature vectors were the cepstra, the delta cepstra and the double delta of the cepstra.

3.4.2 Experiment 1: The Consistency Experiment

Hypothesis 1: The features that predict errors are consistent throughout the corpus.

The hypothesis being tested here is that there is consistency amongst the data; that is, the predictive features from various parts of the data work on all the other parts. If we are to be able to predict whether or not recognitions of new speakers are correct, this type of consistency is essential. For classification to work accurately the predictive features must hold across time and speakers. One way to test for consistency is by way of cross validation; we chose to test on each tenth utterance and train on the balance. By moving the testing utterance one place forward in the corpus each time, ten tests can be made across the entire corpus. We left the corpus in its original utterance order: one speaker after another, each contributing over seventy utterances before the next speaker took a turn. In consequence speakers did contribute to the training set that they were subsequently tested on.

We carried out the experimental procedure outlined in Section 3.3.4. In this case we worked on the whole of the test section of the TIDIGITS Corpus, some 12,000 utterances. We built and tested our ten models, each comprising some 1,200 utterances. The detailed results are shown in Table 3.2. There was very little variation: mean accuracy (that is, the proportion of utterances that were correctly classified as misrecognized or not) was 80.2% with a standard deviation of 1.0% and mean Kappa 35.3% with a standard deviation of 2.4%. These findings confirm considerable consistency across the data. We investigate the causes of the errors further in Section 3.6.6.

It should be remembered that the test section of the TIDIGITS Corpus consists of a relatively large number of utterances by a relatively small number of speakers, and in our experiment there will be many cases where the person whose utterance is being tested is also part of our training set. In the next experiment we look at samples that clearly train on one group of speakers and test on utterances from another.

3.4.3 Experiment 2: The Prediction Experiment

Hypothesis 2: Models trained on one group of speakers will accurately classify the utterances of different speakers.

The hypothesis being tested here is that logistic models trained on one group of speakers can predict errors when classifying the speech of different groups of speakers. Real time performance requires training on one set of speakers and testing on strangers. The acoustic models that are available to us are trained only on adult speech (WSJ or

Test Number	1	2	3	4	5	6	7	8	9	10
Accuracy	78.6%	80.9%	78.7%	81.4%	80.0%	80.4%	81.8%	80.0%	80.4%	79.9%
Kappa	33.5%	35.1%	33.8%	40.1%	34.3%	32.5%	36.6%	33.3%	38.2%	35.4%

Table 3.2: Experiment 1: Tenfold cross validation of the logistic study on the Test Section of the TIDIGITS Corpus.

	Set 1	Set 2
Accuracy	85.38%	85.05%
Kappa	18.59%	16.01%

Table 3.3: Experiment 2: Testing on strangers using logistic regression on parts of the testing section of TIDIGITS.

TIDIGITS). In consequence we excluded children's utterances, and this experiment was run by dividing the adult utterances into two sections, Set 1 and Set 2: each set was made up of 2000 male and 2000 female utterances. The utterances were selected so that no speakers were present in both sets. Logistic models were built for each group and tested on the other group. Thus the results of the experiment explored how the models held when exposed to the utterances of adult strangers.

All the results show levels of Kappa lower than desirable, averaging 17.3% (standard deviation 1.8%),[5] but Table 3.3 shows that predictive accuracy was quite consistent. The average accuracy was 85.2% with a standard deviation of 0.2%. The low levels of Kappa require some explanation. The higher levels of Kappa that were achieved when using the CU Corpus later in this Chapter (Section 3.5.3) may provide a clue as to what could be causing this. A working hypothesis that might account for the difference, is that the TIDIGITS data represents a very refined form of sample with almost no noise in the signal; that is, no side noise, nor any alteration to the sounds arising from their being carried over the phone network. On the other hand the CU Corpus was derived from speech across the phone system. The hypothesis is that it is much easier to pick out non-voiced sounds that are confounding the acoustic models than voiced sounds. Subject to this caveat, the results of Experiment 2 support the hypothesis that models trained on one group of speakers can predict errors when classifying the speech of different groups

[5] It is not normal to provide a baseline when quoting Kappa as Kappa is a measure of distance between the baseline (the agreement that arises by chance) and complete accuracy. However, the agreement that would arise by chance in this experiment is 82.12%.

of speakers and, certainly, we ultimately achieve high levels of accuracy and Kappa when incorporating this technique into our final classifier in Chapter 5.

The fact that Experiment 2 achieved higher accuracy but a lower Kappa than Experiment 1 requires some explanation. Children's speech is poorly recognized when using acoustic models derived solely from adult speech, as is the case with TIDIGITS. Removing the children's data from our corpora, as we did with Experiment 2, reduced the utterance error rate from 19.9% to 1.86%. This resulted in very little data for our logistic models to identify error-prone utterances. In consequence our models were overtrained on correctly recognized utterances and undertrained on error-prone ones.

When predicting that an utterance will be recognized correctly the percentage of correctly recognized utterances in the corpus affects the level of success. This is to be expected. If half of the utterances in a corpora are correctly recognized the probability of correctly predicting which utterances are correctly recognized is 0.5^2 or 0.25. However, if the recognizer is more successful, say it produces the correct hypothesis 90% of the time, the probability of correctly predicting which utterances are correctly recognized is 0.9^2 or 0.81.

In consequence we found ourselves with a model that was trained to spot correctly recognizable utterances and a corpus of almost completely correctly recognized utterances. Hence accuracy improved, but the model did not do as well in classifying the error prone utterances and in consequence Kappa, which takes into account false negatives was poor.

3.4.4 Experiment 3: The Error-Prone Speakers Experiment

Hypothesis 3: Errors are not uniformly distributed over speakers.

Here the hypothesis is that recognition errors are speaker-related. In order to explore the effect on recognition accuracy of different acoustic models we recognized our test set (in this case some 4225 adult male utterances from the TIDIGITS Testing Set) against two different acoustic models: one produced from the TIDIGITS Training Set and another from the WSJ Corpus. The phoneme set produced by the TIDIGITS Training Set is more limited than the one produced by WSJ, which is a very large corpus of read material. We expected fewer errors with the TIDIGITS acoustic models, and that was what occurred. The rate of misrecognized utterances rose from 1.92% to 6.49% when the recognitions were run against the acoustic models derived from a different domain.

The corpus on which the TIDIGITS acoustic models had been trained was also available. We used Sphinx4 to recognize some 4,069 male utterances from this corpus using the two sets of acoustic models, TIDIGITS and WSJ. The results confirmed our findings. Of course, the TIDIGITS acoustic models had been produced by training using this set, and there was almost a perfect recognition result with errors as low as

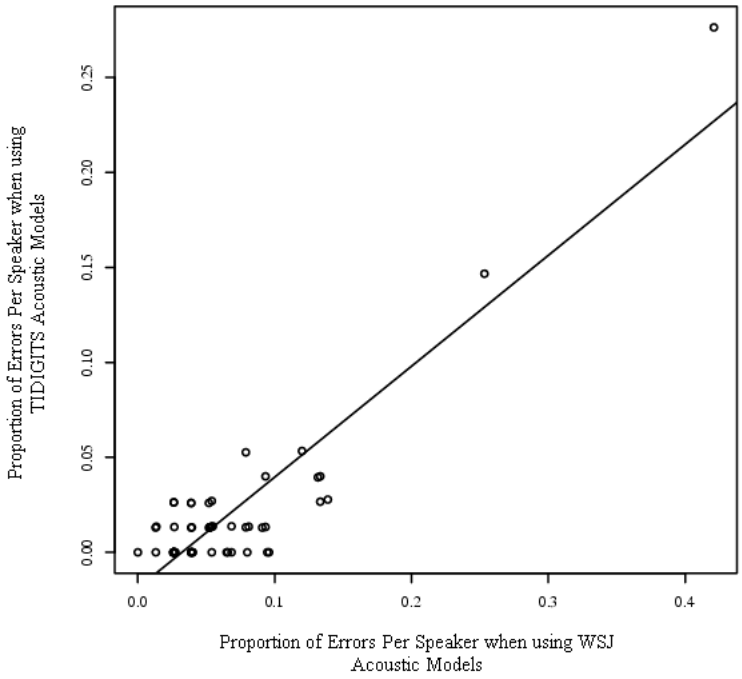

Figure 3.5: Experiment 3: The Error-Prone Speaker Experiment. Proportion of errors by speakers using the TIDIGITS Men Testing Sets with the TIDIGITS and WSJ acoustic models with Sphinx4. Each point is a speaker.

0.39%; but when the WSJ acoustic models were used, the error rate climbed back close to previous experience at 6.12%.

Now we wished to see if some speakers were more error prone than others. Using the data already collected we calculated the proportion of errors per speaker and produced the scatter plot of the proportion of errors shown in the graph at Figure 3.5. We knew that the level of errors should be higher along the WSJ axis than along the TIDIGITS axis. In the experiment it transpired that speakers produced proportions of errors often twice as high when using the WSJ acoustic model. The symmetry of the results using both sets was striking; even though the WSJ model always produced higher proportions of errors, speakers remained as high or low error producers using either acoustic model. In addition, extreme cases with a very high proportion of errors were clear. Listening to the related sound files provided no clues as to why this was occurring. This lead us to conclude that these speakers will face recognition difficulties whenever they speak to a general purpose speech recognizer.

The relationship between errors against both acoustic models is statistically significant since the p-value is less than 0.0001 (p-value: $< 2.2e\text{-}16$) and the regression line explains 81.2% of the variance of the data. Clearly, whatever was causing the problems in the recognitions appeared to cause proportionate problems whichever acoustic model was used, confirming that the cause of these very different levels of proportion of errors by speakers lies with the speaker, not the training sets.

Hypothesis 3 can be tested using Pearson's Chi Square Test for independence. The null hypothesis for the test is that the population proportions of misrecognitions are exactly equal. TIDIGITS was recorded under laboratory conditions so it is very unlikely that channel differences could have affected the sound files. Of course, this is not the case in the CU experiment, where the sound files originate from over-the-phone utterances and channel differences are likely to be present. With TIDIGITS there were a relatively large number of utterances per speaker (just over 70) so the fact that we had two speakers who showed a much higher proportion of errors is unlikely to be due to chance. The results of Pearson's Chi Square Test allowed us to dismiss the null hypothesis; the p-value is very small at 2.2e-6, which is consistent with the hypothesis that the errors are related generally to the speaker rather than to the particular utterance.

3.4.5 Experiment 4: The Goat Experiment

Hypothesis 4: The models once trained will classify which speakers present with recognition problems.

The hypothesis here was that the model produced could predict which speakers would be misrecognized; that is, the logistic function was accurate enough to actually predict the speakers who would generally be recognized correctly or not. The recognition results for the speakers were available from Sphinx4. We produced a predicted outcome for each speaker using the logistic function. We could then plot one against the other. The results are shown in Figure 3.6 and show that the function was good enough to predict speakers who would have higher than average error rates. The relationship between the two variables is significant since the p-value is less than 0.0001 (p-value: $< 6.706e\text{-}14$) and the regression line explains 65.0% of the variance of the data. The graph shows two outliers, and if you exclude them R^2 drops to 30%.

3.4.6 TIDIGITS Experiment Overview

It is clear from our experiments that:

- the acoustic features that reveal errors are consistent throughout this corpus;
- logistic models trained on one group of speakers can predict errors when classifying the speech of different speakers;

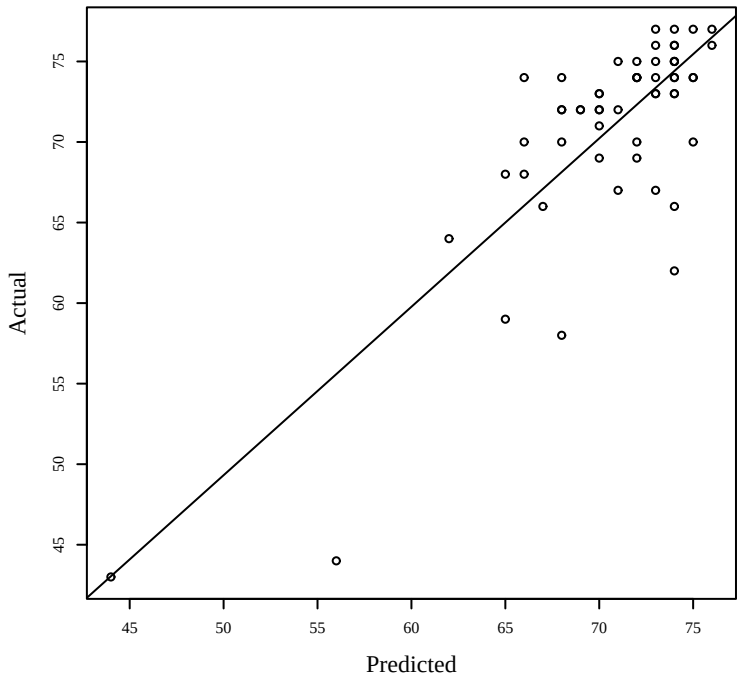

Figure 3.6: Experiment 4: The Goat Experiment. The logistic function's prediction of correct utterances by speaker and the actual outcome for TIDIGITS. Each point is a speaker.

- the level of recognition errors increases when using acoustic models from another domain than the test set;
- the evidence is consistent with the hypothesis that errors are related to some general qualities of the speaker and not of the utterance; and.
- a logistic model can predict which individual speakers will be misrecognized.

The next section reports on a similar set of experiments, but this time on our data sets from the CU Corpus. These experiments were undertaken with a view to confirming the findings on the TIDIGITS Corpus on a somewhat more general set of data. While TIDIGITS affords us the opportunity to study a corpus with no language errors and studio quality sound, CU offers us a wide range of over-the-phone sound with both language and acoustic errors present.

Test Number	1	2	3	4	5	6	7	8	9	10
Accuracy	73.1%	73.9%	74.9%	73.5%	75.9%	73.9%	74.5%	72.9%	73.8%	72.7%
Kappa	39.5%	40.7%	43.4%	39.1%	45.0%	39.9%	42.3%	37.7%	39.3%	39.4%

Table 3.4: Experiment 1: Tenfold cross validation of the logistic study on the CU Corpus.

3.5 The CU Experiments

3.5.1 Introduction

The CU Corpus offered the opportunity to repeat the experiments undertaken in the previous section on TIDIGITS on a large corpus of realistic over-the-phone utterances. Many of these dialogues failed after only one or two turns as users hung up. We worked with a subset of the corpus consisting of dialogues with over 10 user utterances;[6] this set consisted of some 15,600 utterances. It should be remembered that we did not have access to the language model for the CU Corpus. While we can assume it was very large, as the recognitions were performed on an n-gram based wide vocabulary recognizer, we were not in a position to discriminate between acoustic and language errors.

3.5.2 Experiment 1: The Consistency Experiment

Hypothesis 1: The features that predict errors are consistent throughout the corpus.

The hypothesis being tested here was that there was consistency amongst the data. The corpus came with a full set of recognitions, so the first stage of our standard experimental procedure was not required. However, we undertook the other three steps of classifying recognitions as correct or incorrect by comparing them with human annotations, extracting the acoustic features, model building and evaluating. The results are shown in Table 3.4. There was very little variation: mean prediction accuracy of classification of utterances as likely to be recognized correctly or incorrectly was 73.9% with a standard deviation of 1.0% and mean Kappa of 40.6% with a standard deviation of 2.3%. Clearly there was considerable consistency across the data.

[6] As mentioned before, the removal of the shorter dialogues can give rise to the criticism that we might have biased the subset, cutting out all problematic dialogues. However, in this case, the utterance error rate actually increased marginally from 36.65% over the entire original corpus to 37.89% in the subset.

	Set 1	Set 2
Accuracy	77.21%	63.75%
Kappa	33.18%	25.1%

Table 3.5: Experiment 2: Testing on strangers using logistic regression on parts of the CU Corpus.

3.5.3 Experiment 2: The Prediction Experiment

Hypothesis 2: Models trained on one group of speakers will accurately classify the utterances of different speakers.

The hypothesis being tested here was that logistic models trained on one group of speakers could predict errors when classifying utterances of others. Once again we divided our corpus into two sets so we could train on one group of speakers and test on the other. Each set contained 7,500 utterances.

Although there was greater variability, once again there was considerable predictive power as shown in Table 3.5. Mean prediction accuracy was 70.5% with a standard deviation of 9.5% and mean prediction accuracy of Kappa 29.2% with a standard deviation of 5.7%. Once again the results support the hypothesis that models trained on one group of speakers can predict errors when classifying utterances from others. Of course this does not emerge so clearly with this data as we are looking at data that contains both acoustic and language errors. The presence of language errors introduces a further potential source of errors that is not captured by our logistic model.

3.5.4 Experiment 3: The Error-Prone Speakers Experiment

Hypothesis 3: Errors are not uniformly distributed over speakers.

Here the hypothesis is that errors are speaker related. We calculated the actual proportion of recognition errors by speaker. We worked with the existing recognitions for this corpus provided by the University of Colorado and no preliminary work was undertaken along the lines of recognition against different acoustic models. However, we were still in a position to consider the distribution of the proportion of errors across speakers.

Figure 3.7 is a histogram of the proportion of errors by speaker; the distribution appears left skewed. As with the TIDIGITS experiment, we use Pearson's Chi Square Test for independence to test the null hypothesis: that the population proportions are exactly equal. It should be remembered that the data here was by no means as reliable as with TIDIGITS as it was recorded over the phone. However, the result allowed us to dismiss the null hypothesis; the p-value is 2.2e-16, which is consistent with the

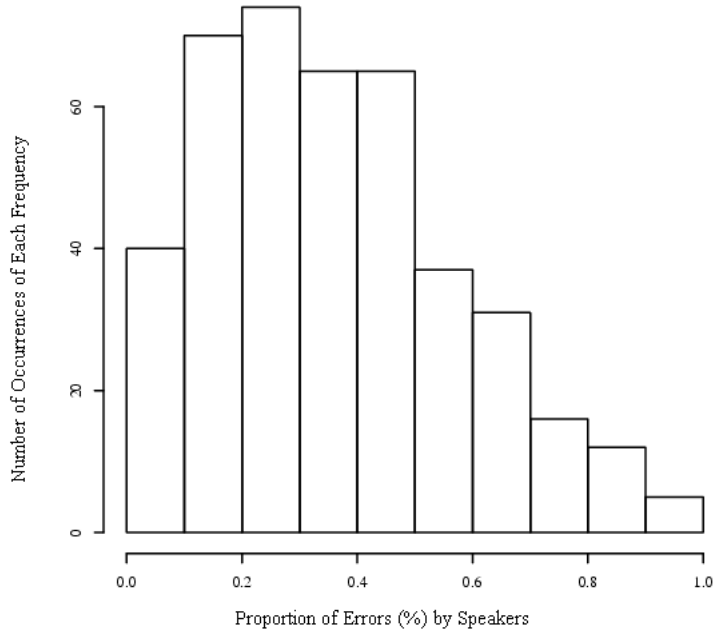

Figure 3.7: Experiment 3: Histogram of the proportion of errors by speaker from the CU Corpus.

hypothesis that the errors are related to the way certain individuals speak.

3.5.5 Experiment 4: The Goat Experiment

Hypothesis 4: The models once trained will classify which speakers present with recognition problems.

The hypothesis here was that the model produced can predict which speakers will be misrecognized; that is, the logistic function is accurate enough to actually predict the speakers who would be recognized correctly or not. The actual outcomes for the speakers were available from the annotations of the corpus and we produced a predicted outcome for each one using the logistic function. The results are shown in Figure 3.8 and show that the function was good enough to classify speakers. The relationship between the two variables is significant since the p-value is less than 0.0001 (p-value: $< 6.706\text{e-}14$). The regression line explains only 49% of the variance of the data, but in this sample we also have errors arising from the language model, which the prediction technique was

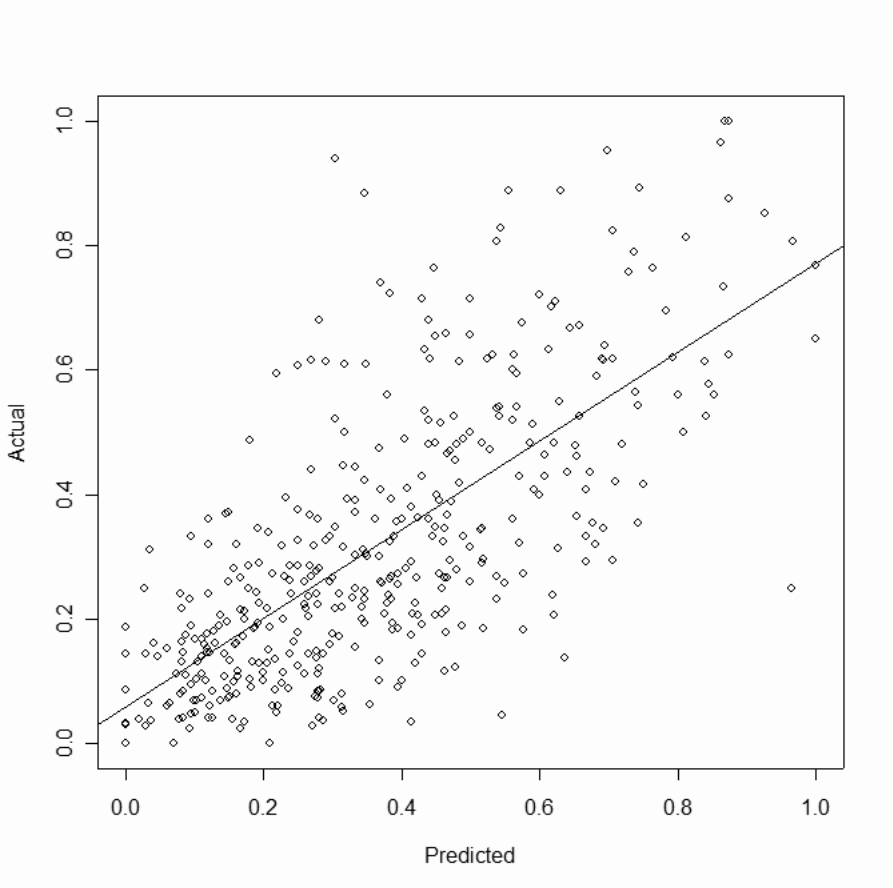

Figure 3.8: Experiment 4: The logistic function's prediction of correct utterances by speaker and the actual outcome for the CU Corpus. Each point is a speaker.

not trained to spot.

3.5.6 CU Experiment Overview

While the presence of language errors in the data complicated matters, the experiments carried out on the data set from the CU Corpus support the results gained from the experiments on the TIDIGITS Corpus. Once again the experiments confirm:

- the features that reveal errors are consistent throughout this corpus;
- logistic models trained on one group of speakers can predict errors when classifying the speech of different groups of speakers;
- errors are related to some general qualities of the speaker and not of the utterance; and
- a logistic model can predict which individual speakers will be misrecognized.

The literature and our work can give some further insights into the causes of these errors. We turn to this aspect in the next section.

3.6 The Causes of Acoustic Errors

3.6.1 Introduction

There is something of a tradition in the literature of looking at the role of hyperarticulation in speech recognition errors (see for example Stifelman [1993], Oviatt et al. [1998] and Levow [1998, 1999]). In the different area of *speaker* recognition, Doddington et al. [1998] show that some people cannot be identified as easily as others. If the same were true for speech recognition it would go against that tradition. That is, it would support the hypothesis that errors arise owing to some general quality or qualities of the way a person speaks, not as a result of the prosodic variations between one utterance by a speaker and another utterance by the same speaker. Such a position is based generally on the view that the distance between the test utterances a particular speaker will produce and the recognizer's training set will cause errors for the reasons explored in this section; prosody should be a minor feature in this problem. In making this comment we are following Huang et al. [2001]'s definition of prosody: the variations between the way an utterance is delivered designed to affect its meaning. One of the problems in the discussion of the differences in the hyperarticulation and sheep and goats positions is that the term 'prosody' is often used loosely, simply to characterize features such as pitch, intensity and timing whatever they are used for or whatever their linguistic effects.

Hirschberg et al. [2004] is a major work in this area; page 163 looks particularly at the issue of hyperarticulation. The features they found that correlate with errors were also associated with hyperarticulation in earlier literature. However, they had hand-annotated their corpus for utterances, that appeared to be hyperarticulated to the human ear. These amounted to some 21% of the corpus. When they excluded the hand-annotated hyperarticulated utterances they still found that the remaining utterances showed essentially the same significant acoustic differences between correct and incorrectly recognized speaker turns. They had not carried out experiments, such as we have reported on in this chapter, that show that some speakers are essentially error-prone. They came to the conclusion that there are some hyperarticulatory trends that the annotators had missed, but the machine still picked up.

We prefer to favor the alternate hypothesis that some speakers present with a high level of errors while some do not, and that in this lies a root cause of errors in the acoustic domain. The results of our experiments support this hypothesis. In order to develop the background to this argument, Section 3.6.2 looks briefly at the source-filter theory of speech production, which is the view of speech underlying automatic speech

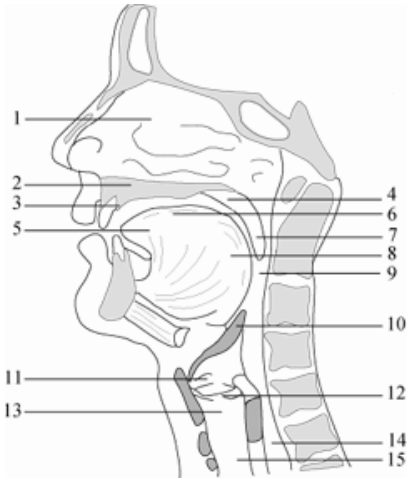

Figure 3.9: The human vocal organs. (1) Nasal cavity, (2) Hard Palate, (3) Alveoral ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Epiglottis, (11) False vocal cords, (12) Vocal cords, (13) Larynx, (14) Esophagus, and (15) Trachea. [Lemmety, 1999].

recognition. Section 3.6.3 reviews how possible it is that prosody is maintained in the acoustic models used by speech recognizers. Section 3.6.4 reports some experimental evidence that supports the view that prosody is not captured in the acoustic models typically used by automatic speech recognizers and, therefore, would not be a significant cause of errors in this domain. Section 3.6.5 reviews the evidence relating to prosody and, finally, Section 3.6.6 moves on to see what evidence of the causes of errors can be found in the logistic models that were produced during our experiments.

3.6.2 The Source-Filter Theory of Speech Production

The human vocal organs as shown in Figure 3.9 produce speech. From a signal-oriented point of view, the production of speech is widely described as a two-stage process [Koreman, 1996]. The source-filter theory of speech production treats the air flow that is forced through the larynx, which contains the vocal cords, as the source of speech and its pitch; and the manipulation that takes place in the three main cavities of the vocal tract, the pharynx and the oral and nasal cavities, as the filter [Fant, 1960; Titze, 1996; Stevens, 1997; Harrington and Cassidy, 2000]. All of this work has its origin in the source-filter theory of speech production proposed by Fant [1960]. This theory assumes that source and filter are independent of each other; but there is some evidence that there is some interaction between the vocal tract and a glottal source. For example, Rothenberg [1981] and Fant [1986] show that the degree to which the vocal tract is open or closed affects the rate of air flow through the larynx. However, these minor

inconsistencies have not stopped Fant's theory of speech production still being widely used, and, in particular, used in speech recognition.

The idea that one might separate the source from the filter proves to be the basis for modern speech recognition across multiple speakers. As Cassidy [2002, pages 35-40] explains, if one were to be able to separate the filter from the source, the filter would provide valuable insights into the disposition of the vocal tract at the time the sound was being made. This would allow one to characterize the features associated with particular sounds emitting from the mouth. As many of the sources of variability in the signal arise from the source, such a separated signal would be easier to recognize across speakers. A common technique now used is to subject the signal to cepstral analysis, but other techniques such as linear predictive coding achieve the same end.

The key point to be taken from this is that modern speech recognizers remove the source (the pitch) from the signal. Pitch is a major component in prosody, but based upon the approach taken by modern speech recognizers, pitch will be largely missing from any acoustic models. However, we will see from Table 3.6 in Section 3.6.6 that pitch is a significant predictor of errors. Maximum pitch is the second most important factor in the TIDIGITS case and tenth most important in the CU Corpus. The literature confirms that pitch is a significant predictive factor for acoustic errors (see, for example, Hirschberg et al. [1999] where maximum pitch is also the most predictive feature). There is an apparent contradiction between the fact that prosody is removed from the recognition process yet pitch, a major component of prosody, remains highly predictive of error prone utterances. This apparent contradiction has to be resolved.

3.6.3 The Production of Acoustic Models

This section looks at how acoustic models are built. It concludes that acoustic models are unlikely to capture prosody when built using current methods. As explained by Huang et al. [2001], prosody is realized by:

pitch, set by the rate of vocal-fold cycling (fundamental frequency or F0) as a function of time;

pauses, that indicate phrases and help the speaker to avoid running out of air;

speaking rate/relative duration, indicated by phone durations, timing, and rhythm; and

loudness, set by relative amplitude/volume.

Obviously models used for recognition can only discriminate the features of an utterance using the very features they record. We will briefly look at how Hidden Markov Models (HMMs) are used in recognition systems in order to assess the chance of these features

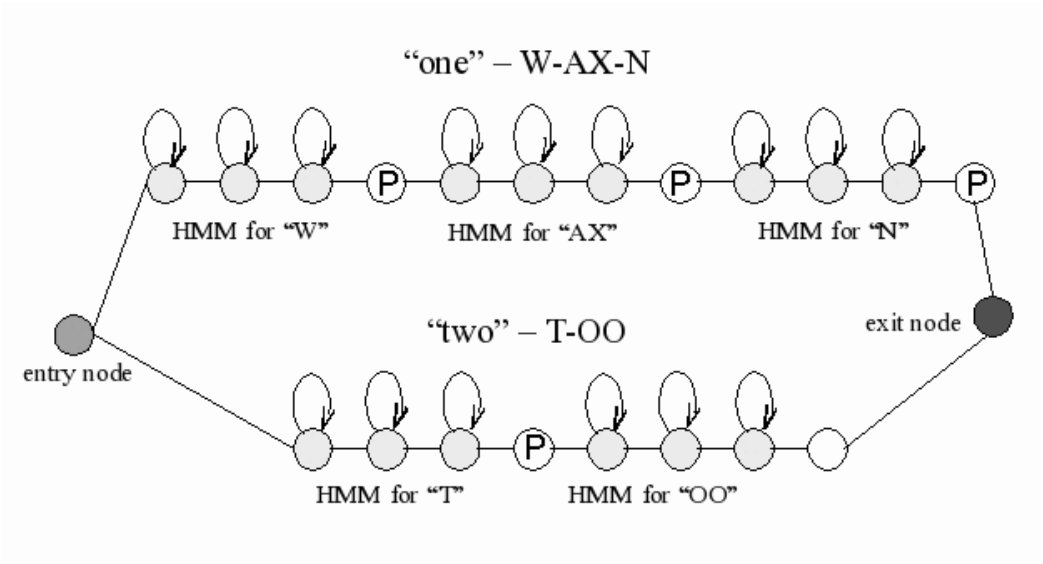

Figure 3.10: Search graph for *one* and *two* using a HMM based graph.

containing information that captures prosody. HMMs are typical for state of the art recognizers and are used by Sphinx4. In the previous section we explained that these models are often based upon cepstral features in order to remove the source effects. This is also the case with Sphinx4; large corpora of spoken language are aligned with annotations and HMMs are trained on them. As Figure 3.10[7] indicates, these models capture the feature vectors associated with phonemes at a sub-phoneme level. For processing, speech is windowed in, say, 20 millisecond segments, so there is plenty of opportunity to model the features associated with the start of a phoneme, its central part and its completion. The HMMs can loop back on themselves to handle timing differences.

Now we can consider the possibility of prosody being captured in the acoustic models. We have already established that the feature extraction process is specifically designed to remove *pitch* (source information). One of the strengths of HMMs is their ability to deal with timing differences; it is, therefore, unlikely that the sorts of *pauses* that play a role in prosody are captured in the models. The HMMs will accommodate and ignore *rate/relative duration* indicated by phone durations, timing, and rhythm. The only prosodic feature left is *intensity* (loudness), but the intensity captured in the acoustic models comes from any part of the training utterances and provides no information in the recognition as to where 'stress' is being applied in the utterance. Given these observations, one must conclude that little prosody could persist from the training set into the acoustic models and, therefore, it cannot be expected to play any significant role

[7]From `http://cmusphinx.sourceforge.net/sphinx4/#acoustic_models`.

in affecting the distance between any speaker's utterance and the recognizer's acoustic models.

3.6.4 Experimental Evidence

In addition to the general analysis of the improbability of prosody being captured in the acoustic models provide above, some experimental evidence emerged during our work supporting the view that prosody is not playing a role in the acoustic models. One would assume that if prosody was captured by acoustic models, those created from corpora richer in prosody might perform differently from those created from corpora more impoverished in prosody.[8]

Sphinx4 is distributed with a number of acoustic models; one is the HUB4[9] model and another the WSJ model. The WSJ model is created from read speech where the original text is articles from the Wall Street Journal. Generally one expects read speech, other than from actors, to be relatively impoverished in prosody. HUB4 is a corpus of actual news broadcasts. One of the qualities of a good news reader is naturalness and one would expect the prosody to be richer, but in any event it is unlikely that the levels of prosody in these corpora would be the same.

As part of the work undertaken in Chapter 4 we had to run recognitions on parts of the CU Corpus using an n-gram language model. Although it turned out to be of no subsequent experimental value, we did so first with the acoustic models for HUB4 and then with acoustic models for WSJ. There was absolutely no difference in the word error rate, which was 44.7% in both cases. If prosody is captured in these acoustic models one would expect there to be some difference in the error rates when using one or the other.

The major point here is that current acoustic models do not seem to perform very differently whatever their prosodic antecedents. Chen and Hasegawa-Johnson [2004] and Hasegawa-Johnson et al. [2005] indicate that prosodically rich acoustic models can be built that do improve recognition accuracy. They created sets of acoustic models and language models that captured and recognized two salient prosody induced acoustic effects: preboundary lengthening and pitch accent tones. They did this by training on a corpus marked up prosodically with TOBI [Silverman et al., 1992].

Intonational phrase boundaries mark intonational boundaries dividing utterances into meaningful 'chunks' of information [Bolinger, 1989]. Wang and Hirschberg [1992] provide us with an example:

(3.2) Bill doesn't drink because he's unhappy.

[8] Considering such a scale of prosody raises the question 'Can one have an utterance without prosody?'. Under Huang et al.'s [2001] definition, one can have utterances that are bereft of prosody. The reading of a series of digits in a flat way would be likely to be such an utterance.

[9] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000S88`.

(3.3) Bill doesn't drink because he's unhappy. (He drinks because he's an alcoholic.)

(3.4) Bill doesn't drink, because he's unhappy. (He believes that alcohol will amplify his depression.)

They explain that if the utterance shown in Example 3.2 is spoken as one sentence, hearers are likely to infer that Bill does drink, but not from unhappiness. This interpretation is expressed in Example 3.3. However, if the utterance is spoken as two phrases as in Example 3.4, hearers are likely to infer that Bill does not drink and the reason for abstaining is his unhappiness.

Pitch accent tones are places in an utterance where a syllable of a specific word has been stressed for a certain semantic effect. For example, the utterance *You mean you don't like chocolate* can be made to convey a number of meanings simply by how it is stressed. It can be said without stress to simply imply its literal meaning, but if *don't* is stressed it implies incredulity.

Chen and Hasegawa-Johnson [2004] and Hasegawa-Johnson et al. [2005] find that the inclusion of prosody does help increase recognition accuracy. Given that no special prosodic training is given when producing current acoustic models, this work confirms that traditionally created acoustic models do not model these features, or only model them in some very weak form. It would appear from this evidence that prosody is not materially affecting levels of errors.

3.6.5 Summary of Prosodic Evidence

In the preceding three subsection we have argued that:

- modern recognizers remove pitch from the signal,
- HMMs remove the effect of timing differences in utterances,
- acoustic models created from large prosodically different corpora do not affect the error rates when used for recognition, and
- acoustic models which incorporate prosody can be created, but are not currently used generally in recognizers, and do affect error rates.

We propose the sum of this evidence supports the proposition that prosody is not playing any material role in misrecognition in current speech recognizers. In the next subsection, we put forward the evidence we have as to the true causes of errors.

3.6.6 The Principal Causes of Error

The logistic function allows the exploration of the effect of changing the relative values of individual predictive features. One can change the value of any particular feature, leaving the others at their mean, and calculate the effect on recognition levels predicted by the logistic model. The features used are those employed in our own experiments and listed at Table 3.1. We adjusted each feature one at a time. It is normal to use the upper quartile and lower quartile figures as appropriate high and low examples of a feature's value. We did this for both the TIDIGITS and CU data and the results are shown in Table 3.6.

All features are normalized before being used in the logistic model. The figures in the range column represent the number of the points change in recognition accuracy between the model using the mean for all the other features and the model using the upper and lower quartile figures for that particular feature. The sign indicates if recognition improved or deteriorated as the feature moved from a low to a high value. Both lists have been ranked by significance, that is the features are listed with the feature with the largest effect first and the lowest effect last. Intensity is the feature that appears at the top of both lists. In the case of TIDIGITS the model indicated that recognition accuracy would be 74.2% with a low intensity figure and 93.8% with a high intensity figure. Interestingly, in the case of CU, intensity worked in the contrary manner: a low intensity produced recognition accuracy of 86.3% and high intensity recognition accuracy of 41.6%. The significance of this is not clear. The measure we used for intensity was decibels. It should be remembered that the recognitions for these two corpora were produced by different recognizers. Acoustic models are designed to match the intensity of cepstral features and the training sets may impart some ambient intensity to such models. Presumably it is just as easy to see errors arising from overshooting that ambient intensity as from undershooting it. However, given the efforts that are taken in modern speech recognition to minimize the affects of simple variations in intensity it is unlikely that these affects would be material. A more likely source of the problem would lie with some common cause once removed that affects intensity in the signal and some other feature(s) that do significantly affect the automatic speech recognition process.

In Table 3.6, we have emboldened the features that appear not to have been explored in the literature as they are associated with formants and voice pathology relating to errors made by speech engines in the acoustic domain. Formants are intimately associated with the shape and length of the vocal tract and voice pathology with the source of speech. Interestingly, these personal physical aspects of the speaker's vocal equipment feature heavily in the list.

We took the more important features (the most significant half) and grouped them according to the phenomena they captured. We ended up with three principal groups:

TIDIGITS	CU
Feature	Feature
meanIntensity	meanIntensity
maximumPitch	**ratioF3ToF1**
maximumIntensity	**ratioF2ToF1**
ratioF2ToF1	**meanF2**
length	maximumIntensity
percentUnvoicedFrames	**meanF3**
ratioF3ToF1	**percentOfVoiceBreaks**
ratioIntensity	minimumIntensity
shimmer	**meanHarmonicity**
numberOfVoiceBreaks	meanPitch
noSyllsIntensity	**percentUnvoicedFrames**
maximumHarmonicity	**meanF5**
percentOfVoiceBreaks	**numberOfVoiceBreaks**
meanF4	length
jitter	noSyllsIntensity
meanF5	**meanF4**
speakingRate	**shimmer**
minimumPitch	**jitter**
minimumIntensity	**minimumHarmonicity**

Table 3.6: Sensitivity of recognition to variation in acoustic factors ranked by significance. Features in bold are those associated with formants and speech pathology.

Intensity, Pitch/Formants and Pathology.[10] We present them in Table 3.7. The arrow after each feature indicates the polarity of the feature; an arrow pointing up means that, as the numerical value of the feature increases, recognition improves, and vice versa.

As already mentioned, in the case of TIDIGITS higher meanIntensity increases errors, but with CU it reduces errors. The very purpose of the acoustic models is to try to capture the intensity of the cepstral features of the training set for comparison with those of the testing set, so the presence of intensity as a major feature is not surprising.

Next we see a number of features associated with the vocal equipment: various formants and pitch. We know the recognizer ignores pitch, but variation in pitch still correlates with errors. This must be pointing to something like difference in the physical size of the entire vocal equipment that could affect recognition. This is reinforced by

[10]Length is only present in the TIDIGITS work.

Group	TIDIGITS Feature	CU Feature
Intensity	meanIntensity ↑ maximumIntensity ↓ ratioIntensity ↑	meanIntensity ↓ maximumIntensity ↑ minimumIntensity ↑
Pitch/Formants	maximumPitch ↓ ratioF2ToF1 ↑ ratioF3ToF1 ↓	 ratioF2ToF1 ↓ ratioF3ToF1 ↑ meanF2 ↑ meanF3 ↓
Length	length ↓	
Pathology	percentUnvoicedFrames ↑ shimmer ↑	percentOfVoiceBreaks ↑ meanHarmonicity ↑

Table 3.7: Groups of significant acoustic factors.

the presence of formants, which also relate to the size and shape of the entire vocal equipment.

Finally, we have various speech pathology features. It is highly unlikely that those who choose speakers for the training sets will have chosen a representative group using speech pathology as a criterion. It may well be that the incidence of unvoiced frames (creakiness), or noise that is often associated with speech pathology conditions, cause problems to the recognition process. We found prediction accuracy reduced from 73.9% to 71.5% and Kappa from 28.6% to 19.0% when these speech pathology features were omitted from the full set of features used for classification on the TIDIGITS Adults Test Section. So it would appear clear that these new sets of features do contribute to errors in the acoustic domain.

3.7 Outcomes

This chapter had two principal aims:

- To demonstrate the extent to which we can determine whether an utterance is likely to be misrecognized purely on the basis of the acoustic features of that utterance.
- To see if we could associate errors with the general way individuals speak (the sheep and goats hypothesis) rather than prosodic variation in the way individuals' articulate utterances from time to time.

As part of the work we had the opportunity to explore which acoustic features cause problems and to look at the effect of some novel features such as those associated with speech pathology.

The chapter looked at the work we did on the TIDIGIT Corpus and the CU Corpus studying acoustic errors. Section 3.2 looked at modeling and classification, concentrating on logistic regression, our principal statistical method. It explained how models can be trained on predictive features, how they can be simplified and improved, and how to assess the prediction accuracy of the classifications these models produce.

Section 3.2 outlined a series of hypotheses:

1. **The Consistency Hypothesis:** Acoustic features that predict errors are consistent throughout a corpus.

2. **The Prediction Hypothesis:** Models trained on one group of speakers will accurately classify the utterances of different speakers.

3. **The Error-Prone Speakers Hypothesis:** Errors are not uniformly distributed over speakers.

4. **The Goat Hypothesis:** Models, once trained, will classify which speakers present with recognition problems.

We know that speech is at least similar enough across different speakers of the same language for them to, normally, understand strangers. Hypotheses 1 and 2 were designed to test if this held true with a machine as the listener. They both tested the reliability of classification of new utterances as likely to be correctly recognized or not based upon experience with past utterances. It is not contentious that recognizers should find it more difficult to recognize utterances if their articulation differs from the articulation in the training set. The question still remained: is it some variation in the way a speaker says things from time to time during the dialogue that causes the problem, or the general way a speaker articulates? Hypotheses 3 and 4 were designed to test the position that errors are generally related to the way individuals speak rather than prosodic aspects of particular utterances.

Experiments designed to test the hypotheses were carried out on both corpora. Section 3.3 shows that with TIDIGITS the data supported all the hypotheses:

1. **The Consistency Experiment** demonstrated that classification was consistent throughout a ten-fold verification: mean prediction accuracy was 80.2% with a standard deviation of 1.0% and mean Kappa 35.3% with a standard deviation of 2.4%. Of course, recordings were studio quality and the language model gave no opportunity for language errors.

2. **The Prediction Experiment** allowed us to train on one group and classify the utterances of strangers, with a mean prediction accuracy 85.2% and standard deviation of 0.2% and mean Kappa of 17.3% and standard deviation of 0.2%.

3. **The Error-Prone Speakers Experiment** allowed us to demonstrate that errors are generally related to some qualities of the way certain individuals speak all the time rather than changes in the way individuals speak from time to time. Pearson's Chi Square Test was used to dismiss the hypothesis that errors were equally spread amongst speakers.

4. **The Goat Experiment** allowed us to classify people who would be misrecognized from the acoustic features of their utterances with a regression line accounting for a significant 65.0% of the variation between the predicted and actual speakers.

As we were not privy to the language models used during recognition with the CU Corpus, we were not able to isolate a set of recognitions that were all covered by that language model and we were unable to operate purely in the acoustic domain. However, Section 3.4 shows that the CU Corpus showed supporting results. The Consistency Experiment reported a mean prediction accuracy of 73.9% with a standard deviation of 1.0% and mean Kappa 40.6% with a standard deviation of 2.3%. The Prediction Experiment reported a mean prediction accuracy of 70.5% with a standard deviation of 9.5% and mean Kappa of 29.2% with a standard deviation of 9.5%. The Error-Prone Speaker Experiment once again confirmed that errors were related to the general nature of utterances issuing from a speaker, rather than some variation within utterances by that speaker; Pearson's Chi Square Test dismissed the hypothesis that errors were equally spread amongst speakers. The Goat Experiment reported a regression line accounting for 49.0% of the variation between the predicted and actual speakers.

Section 3.5 then reviewed the evidence against prosody playing a significant role in causing errors in modern recognizers and put forward the factors we had found that appeared to be the cause of errors. We argued that current acoustic models capture little if any prosody. It appears unlikely that prosody plays a significant part in misrecognition. Indeed, the only prosodic phenomenon dealt with at length in the literature is hyperarticulation. When speech judged by humans to be hyperarticulated was removed from samples by human judges, in Hirschberg et al. [1999], errors could still be predicted, leading one to think some acoustic phenomena other than hyperarticulation is responsible for misrecognition problems. We argued strongly for the sheep and goats hypothesis which our experiments support.

This thesis does not definitely identify the causes of errors in the acoustic domain, but the work does indicate the following underlying causes:

Intensity. Recognizers are sensitive to the general volume levels they face.

Vocal Equipment Size. Pitch and formants reflect the general size of the vocal equipment. Very small, large or malformed equipment will produce utterances not represented in the training set.

Speech Pathology. The HMMs may find it difficult to handle the short breaks in voiced speech that are characteristic of many speech pathology conditions, such as thick, compact vocal cords making contact with the false vocal folds, or the increased levels of noise characteristic of other speech pathology conditions such as lax vocal muscles.

It would appear from these results that the most productive routes to avoid errors lie in discovering why intensity is quite so important given that with one system increased intensity reduced errors yet with the other it increased them, and to produce acoustic training corpora that more fairly represented the types of speakers that systems will encounter. We note the care with which these corpora endeavor to represent various regional accents, but doubt that any corpus endeavors to include a suitable proportion of speakers with colds when, at any one time, a significant proportion of users will have a cold that affects the way they sound.

Returning to the aims of this chapter, we have shown that, when faced with purely acoustic errors, logistic regression can classify utterances from strangers as likely to be misrecognized with around 85% prediction accuracy, and that errors are speaker and not utterance related. In the next chapter we move on to look at the role of language difference in error. This is the problem that arises when users employ words and/or syntax that the recognizer is not expecting.

Chapter 4

Errors in the Language Domain

4.1 Introduction

In Chapter 3, we explored one of the causes of speech recognition errors: difference in the acoustic domain between the model being used by the recognizer and the acoustic features of the utterance being recognized. We demonstrated that one can predict if errors are likely to arise with a reasonable degree of prediction accuracy; that errors are related to particular speakers rather than particular utterances; and that novel features associated with speech pathology and the general nature of the speaker's vocal equipment are playing a significant role in misrecognition. In this chapter, we explore the second cause of errors, differences in the language domain that arise owing to users employing language that has not been modeled in the system's language model. Given the many sources of variability in the acoustic domain, it is essential for a recognizer to employ a language model to guide the search for the correct hypothesis in relation to the utterance being recognized. If an utterance falls outside the language model being employed, in part or in whole, an error occurs as the system cannot return a recognition composed of words or syntax not in its language model. The form this error takes is that the system will return the most likely recognition made up of words in its vocabulary arranged in the patterns allowed by its language model.[1]

Language models are either grammar based or stochastically based. With a grammar-based model, a set of rules is established, covering the words in the model, sometimes their probability of occurring and the patterns they can be arranged in. Stochastic-based models, normally referred to as *n-gram models*, use corpus studies to set probabilities for one word in the model following another word or words. Both types of models are described in more detail in Section 4.2.5. Instances where a recognizer encounters language not in its language model are variously referred to as *out-of-grammar*, *out-*

[1]Sometimes, in the interest of improved recognition speeds, there is such a severe pruning of the search space that there are simply no hypotheses left to offer as a recognition. This matter is referred to again in Section 5.4.3, which reports on using a Nuance 8 recognizer.

of-language or *out-of-vocabulary*. We use the term out-of-language to cover all such cases.[2]

In this chapter we report on the work undertaken to find methods that can detect if an out-of-language utterance is being encountered. Our general approach is to subject each utterance to a second recognition using a looser language model than in an initial recognition.[3] We then use the second recognition to see if we can classify the utterance as out-of-language.

We use three methods. The first method is based upon expanding a grammar-based language model by creating a meta-word that will capture language that is not otherwise part of the defined grammar. The second and third approaches are based upon using a second n-gram-based recognition. In the case of the second approach, the language model is based upon phonemes, and in the third, on the language actually experienced in the domain.

The TIDIGITS corpus offered us the opportunity to work with a corpus that incorporated very high quality sound recordings and a constrained vocabulary: only 11 words. Our working hypothesis was that if we were to be able to separate language errors from acoustic errors we would have to have a dataset that could be completely free of acoustic errors. So we work with that part of the TIDIGITS Corpus that contained no acoustic errors. In order to be able to control the occurrence of language errors we then remove a single word from our language model in order to create out-of-language utterances. We carried out three different sets of experiments using the different secondary recognition approaches in order to assess their performance. Of course, using such a limited language model could run the risk of creating an approach that could not be scaled up. However, we soon found evidence that the approach could scale up and indeed give improve results when scaled up when we compared our first experiment (Section 4.3) with the experiment reported in Kwok [2004]. Using a meta-word to model all the words not in our grammar we achieved an 82.35% accuracy in spotting out-of-language utterances. Working with a more complex grammar Kwok achieved a 93.15% accuracy.

In the remaining sections of this chapter, Section 4.2 looks at the hypothesis we are testing, the methods used and the materials available. Then the next three sections report on the experiments performed with meta-words (Section 4.3), phoneme language models (Section 4.4) and domain language models (Section 4.5). Finally, Section

[2]The expressions out-of-grammar is used to cover language errors in grammar based systems. Strictly, it should be used for cases where absence of a matching syntax is the problem. Out-of-vocabulary is used to cover cases where the word used in an utterance is not present in the vocabulary of the recognizer. Out-of-language is used as a catch all for all these types of problems including absence of suitable n-grams in stochastic based language models.

[3]By looser we mean less constrained; for example, a grammar based language model that contains a path that would match any word is looser than one that does not.

4.6 reports the outcomes. It transpires that meta-words can predict out-of-language utterances with 80–90% prediction accuracy with very little overhead cost; phoneme language models are only accurate to a little better than 50%; and domain language models prove almost completely accurate albeit when using the very specialized data set that TIDIGITS comprises.

4.2 Hypothesis, Methods and Materials

4.2.1 Introduction

In this section we look at the hypothesis, methods and materials we used in our work in the language domain. In Section 4.2.2 we explain the hypothesis our experiments test. All the time we were pursuing ways that we could automatically discover if out-of-language utterances were being encountered.

In Section 4.2.3 we review the materials used in our experiments. Here we used those TIDIGITS utterances that had previously been correctly recognized, but remove one word from our grammar to create out-of-language utterances. In Section 4.2.4 we review the methodology we used, involving running recognitions, using various metrics to classify the results to indicate the presence or absence of an out-of-language utterance, and evaluating the outcomes. In Section 4.2.5 we review grammar-based language models and in Section 4.2.5 we review n-gram based language models.

4.2.2 The Hypothesis Tested

In the language domain we pursued our objective of being able to identify when an out-of-language utterance was encountered by testing the hypothesis:

Comparison with a second recognition derived by using a more general language model will reveal if the initial recognition encountered an out-of-language utterance.

We tested this by carrying out experiments using three different types of more general language models and comparing them with an initial grammar-based recognition:

A grammar model containing a meta-word. This experiment used a recognizer with a further path added to the existing grammar, one that captures any phonemes in any pattern. This is based upon the proposition that if one gives this out-of-language path a very small but discrete probability of being entered, it will pick up out-of-language words, but not match with in-language words.

A phoneme based language model. This experiment used a recognizer with a phoneme based language model based upon the occurrence of phonemes in a very large corpus (WSJ in our case). This is based on the idea of bypassing the fact that hypotheses proposed by the recognizer can only consist of words contained in its

language model. If one relaxes this constraint and only matches at a phoneme level, one may spot sets of phonemes that are out-of-language with respect to the original recognizer's language model. Of course, one then runs the risk that phoneme patterns in unusually spoken in-language words will not be recognized as comprising an in-language word.

A domain language model. This experiment used a recognizer with a language model derived from the actual language encountered in the past by the system, based upon the argument that the best approximation to the language a system will encounter in the future is the language it has encountered in the past. Assuming this produces accurate recognitions for utterances that have been previously encountered, it raises the question of how frequently one encounters new words in practice.

4.2.3 Materials

We worked with the male section of the TIDIGITS corpus which we also used in Chapter 3. However, any utterances that were incorrectly recognized during that work were removed. This meant that we could experiment on a corpus where we experienced no errors prior to reducing the coverage of the language model. There were 3954 utterances in the balance of the corpus we used.

4.2.4 Method

In all experiments, our original set of recognitions, that is those recognitions that are to be classified as containing out-of-language utterances, were produced using Sphinx4. In Chapter 3 the grammar used covered all the digits, *oh* | *zero* to *nine*. Now we remove one digit from the grammar to ensure we encounter out-of-language utterances. The secondary recognition, used to identify out-of-language utterances, is also produced with Sphinx4, but using various different language models as described above. We then use various techniques to characterize the original utterance as out-of-language:

Meta-word If the meta-word is found in the second recognition, the utterance is classified as out-of-language.

Phoneme Language Model An edit distance from the phonemes in the original recognition to the phonemes in the secondary recognition is calculated. We used logistic regression to decide the edit distance above which we classified utterances as in-language and below which we classified utterances as out-of-language.

Domain Language Model If a word is found in the secondary recognition that is not present in the vocabulary available for the first, the utterance is classified as out-of-language.

4.2.5 Language Models

Grammar Based Language Models

Grammar models define both the vocabulary and the allowable syntax of the language expected to be encountered by a recognizer. Example 4.1 shows a grammar file in the JavaScript Grammar Format.[4] It has been edited to give an incomplete coverage of the set of digit sequences.

(4.1)
```
#JSGF V1.0;

grammar TIDIGITS;

public <digits> = ( three | four | five | six | seven | eight | nine | oh | zero )+;
```

Example 4.2 is a real example that demonstrates the pull of the language model in this case, forcing the recognition within the permitted utterances:

(4.2) Said: one two three four
Recognized: oh nine three four

N-gram Language Models

There is a stochastic approach to the production of language models that is based upon the fact that one can capture the probability of the collocation of one word with another by studying large corpora of the language. Such a study will also identify the vocabulary present. Providing one has the corpus in a machine readable form, the process is quick and effective and various software packages exist to carry out the task. Of course, the task of collecting original corpora of spoken language and transcriptions can prove to be expensive.

When looking at large amounts of text, one will find many words such as *boy*, *girl*, *school*, *horse*, and so on following the words *the* or *a*, but one will never find *the ran*. So the fact that a noun may often be preceded by its article is captured purely by counting words, as is the contrary with a verb. These probabilities are expressed as P($horse|the$) = .0000003, or some other small figure. The pairing of two words is called a bigram, three words a trigram and the general approach n-grams.

[4] In the examples shown the variable $< digits >$ can be made up (equals, '=') of one or more (plus, '+') of the words within the brackets. The pipe ('|') allows the words to be alternates. If they had been absent the grammar would only cover all the words in the order shown. Normally the recognizer returns the words specified, but if some information is enclosed in braces ('{}') after a word or words, the recognizer returns that information rather than the word(s).

In theory, if one had enough language data, one could define all the probabilities of all the words over long stretches of language such as ten-grams or 20-grams.[5] In practice, *the sparse data problem* has generally limited the usefulness of n-gram models to the trigram level; these n-gram language models appear to work best when they allow the use of the probabilities for bigrams and even single words, when a trigram cannot be found [Peng, 2001]. This is called 'backing off'; the prediction process moves first to bigrams and then to unigrams. In essence, n-gram language models will allow the recognition of any of the words in their dictionary in any order, but favor the orders that have been seen in the training set. Other approaches used to handle the same problem include *smoothing* [Kawabata and Tamoto, 1996], *discounting* [Zitoumi and Qiru, 2008] and *interpolation* [Joon-Ki and Yung-Hwan, 2006].

In our work we employed the CMU-Cambridge Statistical Language Modeling Toolkit v2 to produce trigram models. This is a suite of tools written in C to assist those working on stochastic language modeling. The basic input is some text file that contains the language that one wishes to model. The final output is a text file, or a binary file containing a language model. Sphinx4 is designed to use these files as one of the various types of language models it can handle.

4.3 The Meta-word Experiments

A meta-word is some sort of constructed 'word' that can match a number of words, perhaps all words, when used in a language model. This section looks at meta-words in the context of grammar-based recognizers. As mentioned in Section 2.5.2, a number of approaches have been attempted in modeling meta-words based upon using phonemes, syllables, morphs and words (see, for example Bazzi and Glass [2000] Bazzi and Glass [2002], Cuayáhuitl and Serridge [2002] and Siivola et al. [2003]). When used to spot out-of-language utterances, the return of the meta-word in the recognition identifies their presence.

One problem with a meta-word is that logically it will also match with the words that are in-language for the recognizer. This problem is managed by allocating a smaller probability to meeting out-of-language material. We looked at one idea that we called Grammar's Complement: if one could construct a meta-word that only matched with words, and perhaps syntax, not modeled by the existing grammar, it could be used directly to find such material and recognition accuracy might increase.

Of course even a superficial consideration of the scope of this problem raises very

[5] With the growth of the Internet such large collections of language may come about. In 2006, Google made the Web IT 5-gram Version 1 available through the LDC. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`. The n-gram counts are generated from 1 trillion word tokens of text; sufficient to usefully observe up to 5-grams.

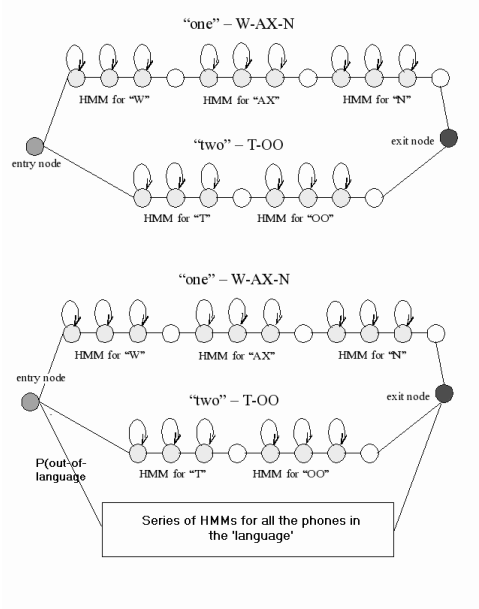

Figure 4.1: Search graph augmented with an out-of-grammar path.

considerable concerns. Not only would one have to be able to create a list of all the words in the language apart from those in one's grammar, but one would also have to specify all the syntactical arrangements and combinations in which the existing words in the grammar could not be used both with and without the additional list of words. Given that there is already published work on the size of the English Language suggesting that the number of words in the language is practically infinite, the task looks impossible [Baayen, 2001; Kornai, 2006].

Given these considerations, we accepted we would have to use a more approximate meta-word, one based on phonemes to limit its size. We accepted that this meta-word would also cover the recognizer's in-language utterances. As mentioned, the technique used to mitigate this is to set a smaller probability on the path leading to the meta-word than on the path leading to the balance of the grammar. We report on the results in this section.

4.3.1 Experiments Undertaken by the Sphinx4 Design Team

The design team on Sphinx4 had looked at the problems of handling out-of-language utterances. Their general approach was to produce a meta-word that included transitions from any one to any other of all the allowable phonemes and side noises. Philip Kwok [Kwok, 2004] undertook the work to add the out-of-grammar path to Sphinx4, basing it on the work of Bazzi and Glass [2000, 2002]. There is work [Hazen and Bazzi, 2001] that indicates the use of a meta-word path is superior to using confidence metrics in picking up such out-of-language utterances. Indeed, at a false acceptance rate of 20%,

Probability of meta-word	Accuracy spotting OOL utterances	Correct OOG	False OOG	Correct IG	False IG
1.00E-60	86.73%	229	6	176	56
1.00E-50	88.87%	239	6	176	46
1.00E-40	91.01%	249	6	176	36
1.00E-30	92.08%	258	10	172	27
1.00E-20	93.15%	266	13	169	19
1.00E-10	92.72%	269	18	164	16

Table 4.1: Kwok's out-of-language test results [From Kwok, 2004], where OOL means out-of-language, OOG means out-of-grammar and IG means in-grammar.

they spotted utterances containing out-of-language words correctly 93.1% of the time against 90.1% with confidence.

Figure 4.1 [6] shows at the top a search graph for the two words *one* and *two*. A further path has been added to the search graph at the bottom of the Figure. Kwok created the path by combining all the context-independent phones[7] in the WSJ set. He reports that recognition accuracy seems to be insensitive to context dependency: when he tried to use phonemes in the contexts in which they could arise in language, it did not increase recognition accuracy. In consequence he simply allowed any pattern of phonemes in his meta-word.

As the meta-word covers all words, including in-language ones, a tunable value for its probability is included and set at some very small value. Kwok tested the results on the an4_words_wordlist test[8] by using a digits grammar to recognize all the utterances in the an4_words test set. This set consists of spoken digits, spelled city names and dates. So in this experiment a grammar that would easily recognize spoken digits such as *one seven two* was also faced with out-of-language utterances such as a spelled city name *p i t t s b u r g h* or a date *march third nineteen twenty eight*. The set has a total of 467 utterances. Kwok obtained the results shown in Table 4.1 when varying the probability of encountering the meta-word. The actual value for probability in the various runs is shown in column one. The last four columns show the utterances correctly classified as out-of-language, incorrectly classified as out-of-language, correctly classified as in-language and incorrectly classified as in-language. The second column

[6] Adapted from `http://cmusphinx.sourceforge.net/Sphinx4/`.

[7] Phones can be realized in different ways depending on the context, that is the nearby phones also being enunciated. Acoustic models can be built that take context into account or ignore it.

[8] This is one of the standard tests sets that is shipped with Sphinx4.

Word Removed	Proportion of total	Word Error Rate
one	9.3%	10.9%
two	9.3%	11.9%
three	9.1%	16.7%
four	9.2%	12.7%
five	9.5%	11.7%
six	9.2%	12.4%
seven	9.2%	13.3%
eight	8.9%	10.8%
nine	9.3%	12.7%
oh	7.5%	7.7%
zero	9.6%	16.9%

Table 4.2: Word error rates when removing single digits.

shows the prediction accuracy with which the technique spotted whether an utterance was out-of-language. The best result is over 93%. Clearly this looks like a technique that can produce very respectable results with a minimum of overhead in terms of additional processing or preparatory work.

4.3.2 Experimenting with a Meta-word on TIDIGITS

We decided to repeat these experiments but to make the task harder than Kwok [2004] in order to see how this affected results. We looked at finding out-of-language utterances containing only one or two words out-of-language rather than the whole utterance being out-of-language, and did so in a set of utterances where around 90% are in-grammar compared with around only 40% in his set. By removing one word from our grammar, say *six*, we would try to spot if an utterance such as *six two three five* was out-of-language. We were using Sphinx4 and could use the same meta-word as Kwok did in his own experiments.

The part of the TIDIGITS Corpus we used for the experiments in this chapter consisted of 12,731 words. The proportion of these made up of each of the eleven digits varied slightly. The mean was 9.1% with a standard deviation of 0.56%. The exact proportions are shown in the second column of Table 4.2. Initially, we omitted each digit in turn from our grammar and ran recognitions and measured the number of errors.

One might expect that the removal of one digit at a time would result in an increase in the word error rate equal to the proportion of words that word represented, but, as Table 4.2 shows, the proportions differed. Clearly, the search for the best recognition

Probability	Accuracy	Correct OOG	False OOG	Correct IG	False IG
1.00E-70	77.57%	156	34	2911	853
1.00E-60	81.99%	464	167	2778	545
1.00E-50	82.27%	524	216	2729	485
1.00E-40	82.35%	601	290	2655	408
1.00E-30	81.01%	660	402	2543	349
1.00E-20	79.34%	715	523	2422	294
1.00E-10	77.16%	762	657	2289	246

Table 4.3: Prediction accuracy of an out-of-grammar path at various levels of probability.

is more complex than simple omission. A incorrect word in one place can have knock-on consequences on the words identified elsewhere in the graph, causing words either side of the out-of-language word to be misrecognized and producing word error rates in some cases nearly twice the actual incidence of the out-of-language word. In these cases the phonemes contained in the missing word have either been matched with the phonemes of the preceding and following words with or without the proposal of a small word between them, or more than one word has been proposed as an alternative to the missing one.

Finally, we ran our experiment by removing the word *one* from our grammar. The error rate when removing *one* was similar to the proportion of words it represented in the corpus and it represented a fairly typical proportion of the digits. We chose *one* rather than, for example *three*, as it minimized the level of error that arose in the recognitions, thus pursuing our goal of making our task harder than the task reported in Kwok [2004]. We ran recognitions and repeatedly changed the probability of entering the meta-word path each time; the full results are shown in Table 4.3. This table follows the same format as Kwok's table. It can be seen that the best result at identifying a single unknown word in an utterance ran at a little over 82%, compared with Kwok at 93.15% for a simpler task. Our classification showed a Kappa of 51.7%, placing it well away from any chance outcome.

4.3.3 Summary

Our work confirms that one can use a path in the grammar comprised of a meta-word modeling any pattern of phonemes and obtain respectable prediction accuracy in identifying out-of-language utterances even in a fairly difficult task: picking out one or two out-of-language words in an utterance. We achieved accuracies of over 80% and

Kwok [2004] achieved over 90% in a simpler task (picking out utterances that were entirely out-of-language). Interestingly, the optimum probabilities reflected this with the single word task being much more improbable at e^{-40} against e^{-20} for the whole utterance task; that is, the optimum level of probability for distinguishing between completely out-of-language utterance and in-language utterances was higher than the optimum probability for distinguishing single words.

Given that this technique can be applied in any system without training, it represents a very useful method for identifying speech recognition errors in the language domain. In the next section we look at the role phoneme language models can play in this task.

4.4 The Phoneme Language Model Experiment

The meta-word we used ignored the phonotactic constraints of language; that is, the patterns of phonemes were not restricted only to those conforming with legitimate patterns of phonemes. Any pattern was allowed. As an English speaker, one knows that *croddle* and *berepot* are possible words, but *cprit* and *bbmlit* are not. There is work such as that of Gallwitz et al. [1996] and Bazzi and Glass [2002] that indicates that the production of meta-words which capture only the legitimate patterns, unlike the meta-word we used in the previous section which captures all patterns, can improve out-of-vocabulary spotting.

One simple way to try to capture the legitimate patterns of phonemes allowed by the language is to undertake a corpus study, to extract the allowable phoneme sequences, and to produce an n-gram language model based upon these phonemes. We took the dictionary derived from the WSJ Corpus. It contains 129,247 words. Each word appears in its lexical form followed by the phonemes that make it up, as shown in Example 4.3.

(4.3) aberrational AE B ER EY SH AH N AH L

We removed the words and were left with a very large corpus of legitimate sequences of phonemes. We used the WSJ dictionary phoneme entries as the input for the CMU-Cambridge Statistical Language Modeling Toolkit [Clarkson and Rosenfeld, 1997]. The result was a language model constructed from phonemes, where the phonotactic constraints of the language were captured by the probabilities of the n-grams.

The hypothesis being pursued in this section is that recognition using only patterns of phonemes that are legitimate in the language will better match those parts of an utterance that are out-of-language than the meta-word used in the last section, where any phoneme could follow any phoneme. For example, TIDGITS contains the utterance *one seven four three zero four four*; when recognized with a grammar that excluded *three* Sphinx4 produced *one seven four four eight zero four four* as a hypothesis. Table 4.4

Utterance	one	seven	four	three		zero	four	four
Word Based	HH W AH N	S EH V AH N	F AO R	F AO R	EY T	Z IH R OW	F AO R	F AO R
Phoneme Based	HH W AY	S EH N AH	F AO R	T R IY		Z EH R OW	F AO R	F AO R

Table 4.4: Comparing a word based and a phoneme-based recognition.

shows the utterance, the phonemes in the word based recognition and the phonemes in a phoneme-based recognition.

In the case of the grammar-based recognition, the absence of the *three* from the grammar has confused the recognition, which has proposed two words *four eight* for this part of the recognition. We have underlined the relevant phoneme. However, the phoneme recognition produced something along the lines of a *three*; again we have underlined the relevant phoneme. The presence of these different sets of phonemes, when the recognitions are compared, identifies an out-of-language section of the utterance. Unfortunately, if one looks at the balance of the recognitions, one can see that the absence of a language model based upon words does introduce other differences in the phonemes selected; for example, the first word *HH W AH N* for *one* becomes *HH W AY*, when the pull of words in the grammar language model is not present. This 'noise' may overpower the benefits of identifying the gross differences in the word *three*. The next section reports on the experiment we carried out to test this.

4.4.1 The Experiment with Phoneme Recognition

We carried out our experiment working with the subset of TIDIGITS from which all misrecognized cases had been removed, as used throughout this chapter. This afforded us the opportunity to work with a set with a minimum of acoustic distance, that is where no acoustic errors were revealed against the acoustic models being used when a full language model was available. In the remaining sections of this chapter, we create out-of-language recognitions by removing the digit *three* from our grammar. This digit is generally considered to be very confusable, as the meta-word experiments confirm. *Three* accounted for 9.1% of the words in the corpus but produced 16.7% of word errors when removed.

We ran recognitions using this grammar and the phoneme language model. We then compared the similarity of the two recognitions at a phoneme level. We use a similarity metric based upon the idea of edit distance [Levenshtein, 1966] where the permitted operations are insertion, deletion, and substitution. This is easiest to explain with a

character string example. The string *vintner* can be edited to become *writers* as shown in Table 4.5.

S	I	M	D	M	D	M	M	I
v		i	n	t	n	e	r	
w	r	i		t		e	r	s

Table 4.5: Example of edit operations.

Here I, D, and S mean insertion, deletion, and substitution and M means match. As can be seen, there are five operations so the distance between these strings is five. In the example, we had five operations and the word is seven letters long. We take the number of operations from the total number of letters (in this case $7 - 5 = 2$) and then take that figure as a proportion of the number of letters ($2/7 = .286$). So the similarity of these two sets is .286.[9]

We applied the same approach using phonemes as our basic units. If the phonemes in the recognition produced by the phoneme language model are the same as the phonemes in the grammar-based recognition, similarity is 1. The larger the difference, the smaller the similarity, and if the number of edit operations equals or exceeds the number of phonemes in the original grammar recognition, the similarity is 0. The distribution of similarity over the utterances is shown in Figure 4.2. The set of utterances totalled just under 4,000. As can be seen, there is some distance in most of the comparisons.

Previously, we were concerned that the outcome would be confounded by the 'noise' introduced owing to the lack of a word-based language model. This turned out to be the case, but the confusion did not overwhelm the method. We built a logistic model to see how well the similarity metric predicted out-of-language utterances and trained it on the first 90% of our data. This allowed us to classify the balance with a prediction accuracy of 53.7% with a Kappa of 19.7%. The results are shown in Table 4.6.

Another way of looking at the data is to see how accurately the similarity metric predicts errors at various breakpoints. That is, if you assume everything over, say, a 50% similarity is in-language and everything below is out-of-language, how many false positives or negatives will arise? Table 4.7 shows the results across a range of breakpoints. At the best Kappa score (18.3%), which is located at the 60% breakpoint, a prediction accuracy of 55.1% is achieved, confirming the logistic work.

[9]Where the first calculation results in a negative figure, the similarity is treated as being zero.

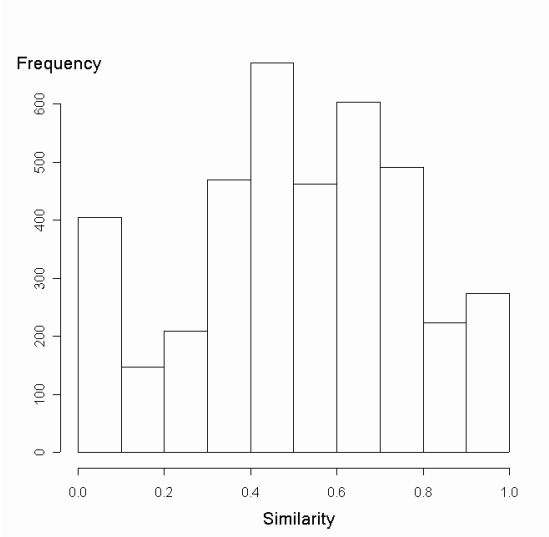

Figure 4.2: Histogram of similarity from the phoneme language model experiment.

	Actually Out	Actually In
Predicted Out	871	1709
Predicted In	120	1254

Table 4.6: Using a phoneme language model to classify out-of-language utterances on the TIDIGITS Corpus.

4.4.2 Summary

Clearly the technique explored here does identify some of the out-of-language utterances, but the noise introduced by the generally looser recognitions provided by the phoneme language model makes the process less precise, with accuracies in the 50% range. The Kappa is low, but tossing a coin would produce a zero kappa. This relatively low prediction accuracy is not surprising as the introduction of word-level models in the recognition process was designed to improve recognition accuracy. Given that this method only requires the building of the phoneme language model once before it can be used in different domains, it does have some attractions, but the general level of prediction accuracy is inferior to the use of a meta-word.

4.5 The Domain Language Model Experiment

In this section we report on the experiment carried out using a potentially very large language model, which we refer to as a 'domain language model'. It is of course the case

Similarity breakpoint	40%	50%	60%	70%	80%
True positives	2097	1660	1390	930	493
True negatives	365	599	791	934	988
False positives	626	392	200	57	3
False negatives	866	1303	1573	2033	2470
Proportion correctly classified	62.2%	57.1%	55.1%	47.1%	37.4%
Kappa	7.0%	12.6%	18.3%	15.4%	8.9%

Table 4.7: Proportion of utterances correctly classified at different similarity levels

that we use only a fraction of the words in the language in any particular dialogue, and there is a well established principle that past language used in a domain is likely to be the best guide to future language that will be encountered in that domain. Increasing the basic grammar of any grammar-based system to include off-topic utterances reduces recognition accuracy in relation to those utterance that are important to the dialogue. However, an n-gram based domain language model can be constructed from annotations of the actual dialogues from any particular domain and to provide a secondary recognition used to identify out-of-language utterances encountered by the grammar-based system.

We put forward evidence to support the contention that past experience is a very good guide to future experience in Section 4.5.2. The literature supports this contention. At page 15, Reynaert [1963] provides a brief summary of the literature including what is generally considered the most significant work in the field by George Kingsley Zipf (see for example Zipf [1935]). To quote Zipf at pages 11–12:

> In any extensive sample of connected English, it will, in all probability, be found that the most frequent word in the sample will occur on the average once in approximately every 10 words, the second most frequent word once in every 20 words, the third most frequent word once in every 30 words, the 100th most frequent word once in every 1000 words, the nth most frequent word once in every 10n words; in brief, the distribution of English approximates with remarkable precision to an harmonic series.

Zipf's Law has been subject to much academic investigation and, while some work supports the view that it is not strictly accurate, Ha et al. [2002] shows evidence supporting the Law over several languages and also reviews the past investigations. One thing that is clear from all the literature is it is generally accepted that some broadly power based relationship exists in the way new words are encountered in language. On this basis it seemed useful to see if we could exploit the Law.

In Section 4.5.3, we model the rate of introduction of new words mathematically to allow us to calculate the required size of training corpora for any given application and other useful variables.

First we report on our experiment where the hypothesis being tested is that analysis of a recognition using a language model based upon domain experience will reveal if an utterance is out-of-language for a particular grammar. The presence of one or more out-of-language words in the recognition is used to make this decision.

Of course, the use of a domain language model to provide a recognition raises the question: why not simply use such an approach for one's initial recognition? However, in practice the extraction of semantic information is considerably easier in grammar-based systems which form the basis of most currently fielded commercial systems.

4.5.1 Experiment with a Domain Language Model

TIDIGITS is a relatively simple domain where all the utterances consist of spoken digits. We created a domain language model from the annotations accompanying some 4,000 utterances. We took the annotations and used them with the CMU-Cambridge Statistical Language Modeling Toolkit to produce a trigram language model [Clarkson and Rosenfeld, 1997].

When we ran a test recognition on the part of the TIDIGITS Corpus being used in this chapter, we found we achieved a recognition accuracy of 99.47%. Clearly, in this rather simple domain, a secondary recognition would reveal any out-of-language word not present in the grammar owing to the very high level of recognition accuracy of the secondary recognition. For example, if *three* were missing from the original grammar based recognition, the second recognition, based upon the domain language model, would almost certainly recognize the word correctly and, therefore, classify the first recognition as having encountered an out-of-language utterance.

While the technique performed well in this case, it raises the question of what level of coverage past utterances in the domain will provide with respect to future utterances. In the next two sections we look at how well this hypothesis holds in the real world. In Section 4.5.2 we undertake two corpus studies and in Section 4.5.3 we look at the possibility of building a mathematical model that fits the data revealed by these studies.

4.5.2 Empirical Studies of Domain Language Models

Stochastic models will back off to unigrams where no bigram or trigram is available to match the pattern of words being handled. This means that they will, in effect, recognize any word in their vocabulary. Therefore, to investigate if they provide coverage of new utterances, we have to investigate the rate at which new words appear in utterances. We looked at two corpora: the Pizza Corpus, where we have a collection of dialogues

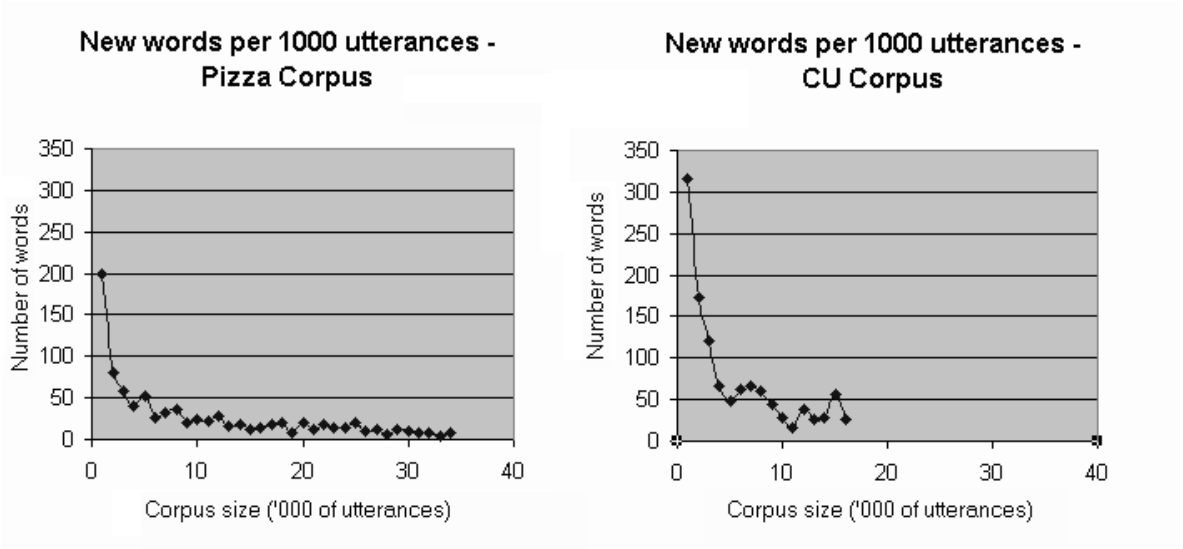

Figure 4.3: Rate of introduction of new words in two corpora.

concerning the ordering of pizzas, and the CU Corpus, where we have a collection of dialogues concerning air travel information.

We took the transcriptions of the dialogues in the Pizza Corpus and analyzed them. There were some 35,000 utterances comprised of only 921 unique words. The chart to the left of Figure 4.3 shows the rate of new words seen per 1,000 utterances. It is clear the chance of seeing unknown words in utterances rapidly reduces, so that when one has the experience of 5,000 utterances it is low, at between 50 and 100 per 1,000; and at 10,000 it settles down at under 50 per 1,000, reducing to more like 10 per 1,000.

Clearly, within the pizza domain a fairly small corpus will provide a quite reliable guide as to future experience in the domain. Of course, any such empirical finding is domain specific and, even though the literature supports the view that these sorts of relationships hold right across the language [Zipf, 1935], we decided to see if same was true in a second domain. This time we used the CU Corpus of air travel information dialogues. We used the same set of longer dialogues that we have used throughout this thesis. They amount to some 16,400 utterances, with a vocabulary of 1,182 words. The results are shown to the right of Figure 4.3. Although there does appear to be a higher incidence of new words, the graphs are quite similar. Indeed, we found it possible to fit a power-based trendline in both cases. The fit was very good with the line accounting for 85.85% of the variation in the data in the Pizza case and 80.81% in the CU case.

Let y be the number of previously unseen words in each thousand utterances and x be the corpus size in thousands of utterances. Then the power-based relationships take the form:

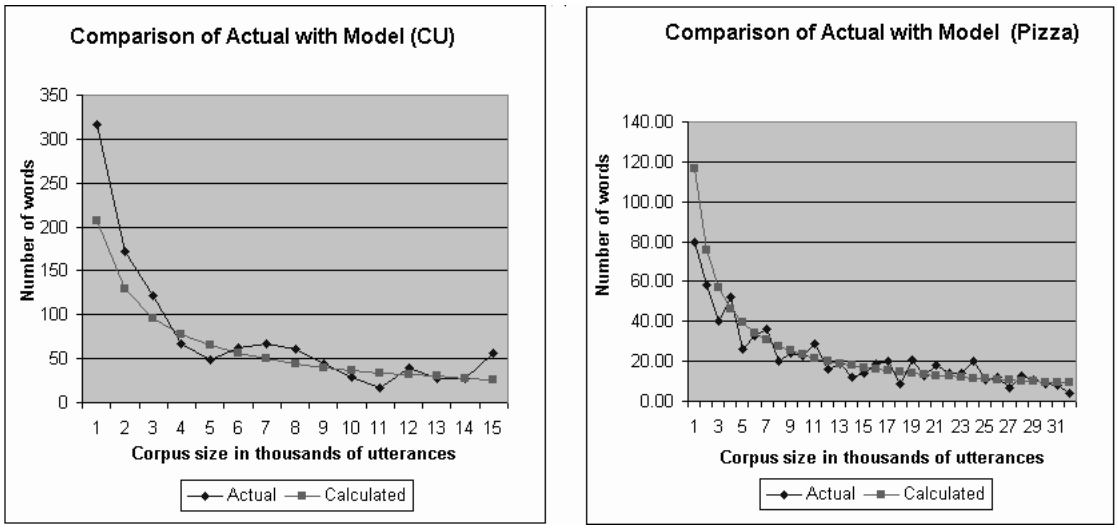

Figure 4.4: Comparison of model with actual out-of-vocabulary words.

$$y = \alpha x^{\beta} \tag{4.4}$$

The actual formula for the Pizza Corpus is:

$$y = 157.73x^{-0.8159} \tag{4.5}$$

and for the CU Corpus:

$$y = 284.61x^{-0.8688} \tag{4.6}$$

The coefficient α was larger in the case of the corpus with a larger vocabulary, but the power β was very similar in both cases.

4.5.3 Mathematical Modeling of Domain Language Models

Müller et al. [1995] show a method based upon set theory to build a mathematical model that can predict out-of-vocabulary rates and the required corpus size to achieve target levels of recognition accuracy, but the previous section revealed that the rate that new words were introduced into a corpus can be described by the formula:

$$y = \alpha x^{\beta} \tag{4.7}$$

where y is the number of new words per 1000 utterances, and x is the size of the training set. We decided to investigate this relationship. Our observations appear to support the view that there is a fairly straightforward power-based relationship between α and β and:

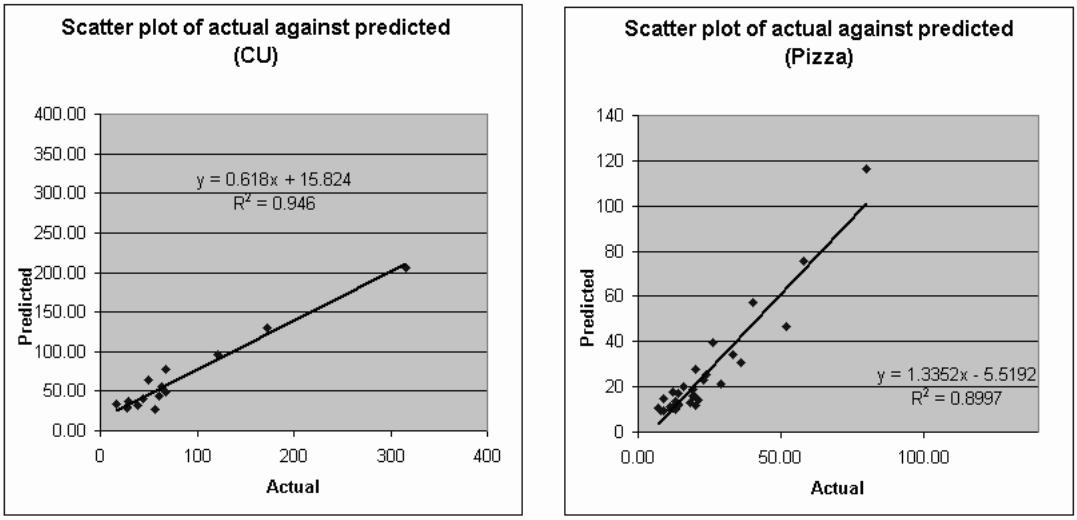

Figure 4.5: Scatter plot of model with actual out-of-vocabulary words.

C_{train} – The size of the existing corpus; that is, the number of utterances available to draw the vocabulary from.

C_{test} – The size of the unseen corpus; that is, the number of utterances one expects to encounter.

$OOV(C_{test})$ – The number of out-of-vocabulary words in the unseen corpus.

This should allow us to model the various features of this problem using integration:

$$OOV(C_{test}) = \frac{\int_{C_{test}}^{C_{train}} \alpha(C_{test} - C_{train})^{\beta}.d(C_{test} - C_{train})}{C_{test} - C_{train}} \tag{4.8}$$

Using Equation 4.9 we can predict the introduction of out-of-vocabulary words:

$$\int_{C_{test}}^{C_{train}} \alpha(C_{test} - C_{train})^{\beta}.d(C_{test} - C_{train}) \tag{4.9}$$

The predictions work very well as indicated by Figure 4.4. In addition, it is clear from the scatter plot of the agreement of the predictions with the actual outcomes (see Figure 4.5) that they achieve a relatively high degree of accuracy; using linear regression they show that the trendline in both cases accounts for 90–95% of the variance in the data.

Of course, the equation can be rearranged to find any missing variable given the others are known, allowing one, for example, to estimate the size of training set required for a given level of out-of-vocabulary incident. This leaves outstanding the question of the size of the parameters α and β for any given corpus. In our two corpora β is

very similar in both cases at a little over -.8, and α does bear some relationship to the relative complexity of the vocabulary in each corpus. Pizza uses 921 words over 35,000 utterances, while CU uses 1182 words over 16,400 utterances; α is 157.73 and 284.61 respectively. Determining whether or not these parameters generalize to other domains constitutes future work.

4.5.4 Summary

In this section we carried out an experiment involving a secondary recognition using a domain language model. It showed that this secondary recognition could identify out-of-language utterances encountered by an initial grammar-based recognition in almost every case.

The main weakness with the domain language model is that it cannot, by definition, contain words that have not yet been encountered. We looked at two corpora and the rate at which new words are encountered. It appeared that, at least in two different domains, providing the training set for the domain language model exceeded 10,000 utterances, new words were only going to be encountered at a rate of less than 25 words per 1,000 words.

Finally, we looked at the possibility of being able to model the rate of introduction of new words into dialogues based upon the formula:

$$y = \alpha x^{\beta} \tag{4.10}$$

where y is the number of new words per 1000 utterances, and x is the size of the training set. This formula appears to fit the data very well and allows us to use integration to estimate such outcomes as the size of a training set for a given rate of new out-of-language utterances in future.

The use of a domain language model proved to be the most accurate of our methods, but this approach is expensive. It requires human annotation of a sizable recorded corpus for the domain in question.

4.6 Outcomes

In this chapter we explored the potential to spot out-of-language utterance, recognized by a grammar-based recognizer. The approach we pursued was to make a second recognition using a looser language model than used in the first instance. We used three variations on this theme and three different clues as to the presence of out-of-language utterances. They are summarized in Table 4.8 and explained below.

Meta-words work by placing a generic word that can model any word into a grammar and tuning its probability to maximize its effect. The presence of the meta-word in

Approach	Description	Clue
Meta-word	Placing a meta-word path in the grammar that can recognize any utterance	Presence of the meta-word in the recognition
Phoneme Language Model	Modeling language at a phoneme level	Edit distance
Domain Language Model	Creating a language model based upon experience in the domain	Presence of out-of-vocabulary word(s)

Table 4.8: The three out-of-language identification methods examined.

the recognition is taken to indicate an out-of-language utterance. **Phoneme language models** model language at a phoneme level. In theory they will recognize the phonemes in most utterances. A large edit distance between the phonemes in the phoneme language model recognition and the phonemes that make up an initial grammar based word level recognition is taken to indicate the presence of an out-of-language utterance. A **domain language model** is based upon the actual language experienced in the past by the system. The presence of a word in the domain language model recognition that is not available in the original grammar based recognizer's vocabulary is taken to indicate the presence of an out-of-language utterance.

Our hypotheses was that *a second recognition using a more general language model would reveal if the initial recognition encountered an out-of-language utterance.* In Section 4.2 we reviewed this hypothesis and the materials used and method adopted in our experiments. Finally, that section reviewed the two sorts of language models used: grammar and n-gram based.

The next three sections reported on the experiments with meta-words (Section 4.3), phoneme trigram language models (Section 4.4) and domain language models (Section 4.5). Each method had advantages and drawbacks. Table 4.9 lists them and we review them below.

The **meta-word** offers an effective and robust out-of-language identification method. It can be created without the use of large resources in advance, and is capable of being used in every domain. Prediction accuracy, that is the proportion of utterances correctly classified as either in or out-of-language, was good at 80–90%. **Phoneme language models** share the quality of being created in advance. The cost of creation was not great, but prediction accuracy was low at around 55%. There is work that sug-

Method	Advantages	Drawbacks
Meta-word	No training No corpus annotation	80 – 90% accurate
Phoneme Language Model	No training No corpus annotation	55% accurate
Domain Language Model	Virtually 100% accurate	Corpus annotation Corpus collection Training

Table 4.9: The advantages and drawbacks of the out-of-language identification methods examined.

gests that 5-grams are the most successful basis for a phoneme based language model [Decadt et al., 2002a]. While the CMU-Cambridge Toolkit, the tool we used to create language models, could create n-gram models of this size, the software to convert them into models to be used with Sphinx4 had a ceiling of trigrams. The **domain language model** offered pinpoint prediction accuracy, but our experiments were in a very narrow domain, spoken digits, and in a real world situation considerable expense would be incurred in the corpus collection and annotation required to implement the method.

In the next chapter we report on a series of experiments we undertook to bring together the outcomes of our work in the language domain reported on in this chapter and in the acoustic domain reported on in the previous chapter.

Chapter 5

Experiments in Classification

5.1 Introduction

This work has divided the causes of errors into an acoustic part and a language part. Chapter 3 explored the causes of errors in the acoustic domain and Chapter 4 in the language domain. In this chapter, pursuing the aim to produce a methodology that can predict the presence of errors with a high degree of certainty, we first experiment using logistic regression on all the features in both the acoustic and language domains. Then we add to the features the outputs from the automatic speech recognizer, such as acoustic distance, normally used in confidence calculations. Next we explore the performance of six machine learning techniques on the data. Finally we evaluate these techniques by re-running these experiments first on a different data set, then using a different recognizer, and then compare these results with the best results in the published literature.

It is useful to summarize the scenario that such a classifier would be used in. Currently there are a growing number of commercial systems fielded where users phone in to complete some task-oriented dialogue, from ordering a pizza to effecting a banking transaction. These dialogues involve the giving of instructions or information to the system by the user using speech. At each dialogue state it is very important to know whether the recognition is correct or contains errors.

Figure 5.1 shows how the classifier and the other software components required, additional recognizers and an acoustic feature extractor, would be integrated with an existing system together with the flow of data through the enhanced system. Sound arrives at a microphone and is digitized. It is then passed simultaneously to four recognizers and an acoustic feature extractor. The recognizers are the original grammar based recognizer and three additional recognizers: one grammar based using the out-of-grammar meta-word path, and two n-gram based using a phoneme and domain language model respectively. The outputs of all these components are passed to the classifier which adds to the original recognition hypothesis a true/false tag.

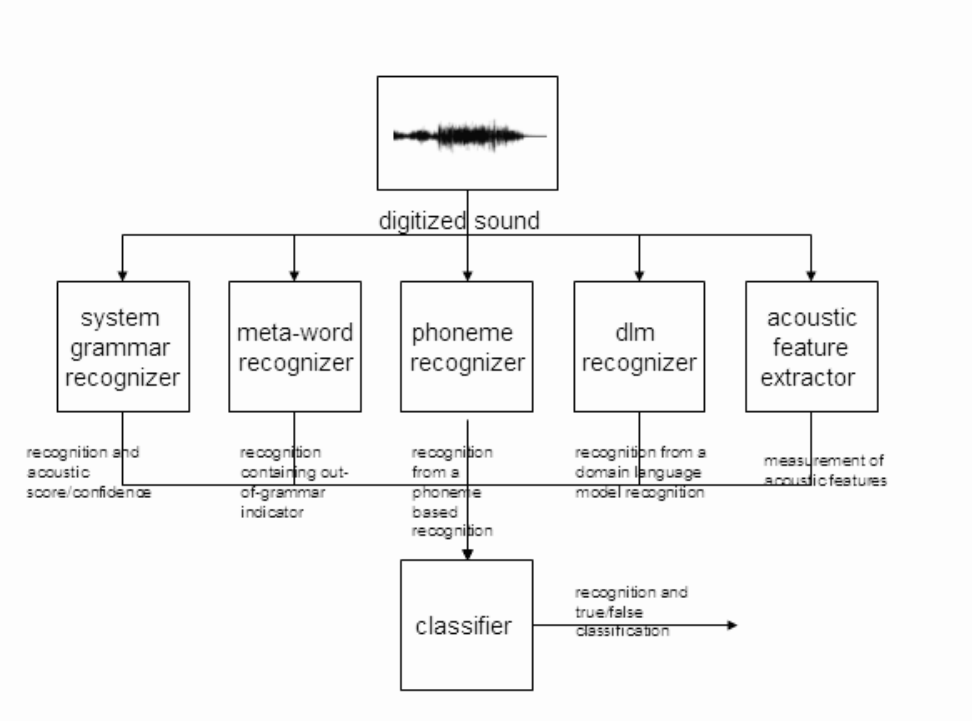

Figure 5.1: Data flow through a unified classifier.

Once implemented, the system would be aware of the advice that the recognition was correct, or incorrect, in much the same way it is currently aware of some confidence index associated with the recognition. This information would be used to assist in the management of the dialogue, controlling the re-asking of questions, or the seeking of clarifications.

The importance of a high degree of certainty in knowing if a recognition is correct or not should not be underestimated. Clarification sub-dialogues appear to cause additional problems when they have to be used in human-machine interactions (Shin et al. [2002], Choularton and Dale [2004] and others). Avoiding unnecessary clarification sub-dialogues is therefore obviously desirable. Only being able to classify recognitions correctly 85% of the time results in over one in seven recognitions containing errors being accepted or correct ones being rejected. Raising prediction accuracy to over 95% reduces this to one in twenty and very significantly reduces the problem of unnecessary clarifications.

Example 5.1, from the Pizza Corpus, shows the kind of problem to which confirmations give rise. Here the system is unsure whether it has heard the user correctly and it prompts with an indirect confirmation at line 4. The result, at line 5, is a protest from the user that is presumably out-of-grammar and the dialogue falters. If the system had greater confidence it had misheard the utterance at line 2 it could have asked the more

direct question "Sorry I didn't get that, can you repeat what you would like for your third pizza?".

(5.1)
1. Prompt: What would you like for your third pizza?
2. Said: a large house @hes@ [fragment] ham and pine hawaiian
3. Heard: @hes@ large house supreme minus olives [confidence: 46]
4. Prompt: o.k. a large house supreme but no black olives
5. Said: incorrect no
6. Heard:[rejected] [confidence: 40]

5.1.1 The Organization of this Chapter

Section 5.2 reports on our experiments using logistic regression over the wider range of features. First, Section 5.2.1 reports on experiments using logistic regression on all the predictive features in both the acoustic and language domains. Section 5.2.2 looks at the automatic speech recognizer outputs. Section 5.2.3 reports on a repetition of the experiments with the additional feature of acoustic distance, an output from the automatic speech recognizer which is normally used in confidence calculations.

Section 5.3 builds on this preparatory work as a benchmark and reports on a series of experiments using six different machine learning techniques: bagging, boosting, naive Bayes, neural networks, Random Forest and support vector machines.

Section 5.4 goes on to evaluate the techniques on real life data from the CU Corpus using both a Sphinx4 and a Nuance 8 recognizer and, in Section 5.4.4, we compare these results with the best results in the published literature [Hirschberg et al., 2004].

Finally, Section 5.5 reports on the outcomes of the work to produce a unified classifier. First, however, in this section we explain the methodology used in this chapter and discuss classification.

5.1.2 Methodology

For the work covered by this chapter we required a general data set that contained both acoustic and language errors. We continued to work with TIDIGITS and made two data sets of 4000 utterances each. The first of these consisted of the first 2000 male utterances and the first 2000 female utterances in the TIDIGITS test set, and the second consisted of the second 2000 male and 2000 female utterances. This approach meant we could train on a mixture of male and female speakers and test on another set of different male and female speakers. This fitted in with the objective of being able to classify the utterances of strangers when they used spoken language systems. We created a class of out-of-grammar utterances by removing the word *three* from the grammar.

We commenced the experiments using logistic regression in much the same way as in Chapter 3: we prepared two sets of utterances, extracted measurements for the various features used in Chapter 3 and Chapter 4 (as reproduced in Table 5.1), and classified the actual outcomes of recognition as correct or incorrect. We trained a logistic model on one set, used AIC to reduce the features and then ran classifications on the other set. Then we added outputs from the speech recognizer and repeated the experiments. Then we moved on to experiment with six other machine learning techniques: bagging, boosting, naive Bayes, neural networks, Random Forest and support vector machines. Finally, we experimented with a different data set consisting of utterances from the CU Corpus, and then with a different recognizer (Nuance 8).

5.1.3 Classification

There are many areas where classification is extremely useful. These range from weather forecasting to facial identification, and, indeed, speech recognition. Some of the most cutting edge work is being done in genetics to classify sequences in the genetic code (see, for example, Gentleman et al. [2005]). Classification is achieved by using models composed of mathematical or formal expressions of the relationships between observed features and outcomes. Such work is referred to as *machine learning*.

We have used logistic regression, which is one technique amongst a group known as generalized linear models, as our principal classification technique. One of the advantages of logistic regression is that it affords the user greater analytical features than many other techniques. In relation to the work in the acoustic domain we were particularly interested in exploring the causes of errors. However, the *No Free Lunch Theorem* [Duda et al., 2001, Section 9.2.1] [Wolpert and MacReady, 1997] points out that there are no context-independent or usage-independent reasons to favor one learning method over another. If one performs better than another, it is owing to its better fit to the particular problem, not its general superiority. Logistic regression is a well established approach where one wishes not only to classify, but also to analyze. It is most useful for this latter aspect owing to the manner in which it determines the significance of individual factors in the model and allows the researcher access to its internal workings. However, there are other techniques that might prove better at classification. Therefore, in this chapter we explore a number of techniques.

5.1.4 Our Approach

There is a considerable literature that addresses issues in machine learning such as the selection of particular methods, the proper handling of data and separation of datasets for the purposes of training, development and testing (see Ripley [1996], Vapnik [1998], Duda et al. [2001], Hastie et al. [2001] and Schölkopf and Smola [2001] for extensive

Feature	Description
INTENSITY	
maximumIntensity	The maximum level of intensity achieved during the utterance in decibels.
meanIntensity	The mean level of intensity achieved during the utterance in decibels.
minimumIntensity	The minimum level of intensity achieved during the utterance in decibels.
ratioIntensity	The ratio of maximum intensity to minimum intensity.
PITCH/FORMANTS	
maximumPitch	The maximum pitch (F0) during the utterance in Hertz.
meanF1	The mean of the first formant in Hertz.
meanF2	The mean of the second formant in Hertz.
meanF3	The mean of the third formant in Hertz.
meanF4	The mean of the fourth formant in Hertz.
meanF5	The mean of the fifth formant in Hertz.
meanPitch	The mean pitch (F0) during the utterance in Hertz.
minimumPitch	The minimum pitch (F0) during the utterance in Hertz.
ratioF2toF1	The ratio of the second formant to the first formant.
ratioF3ToF1	The ratio of the third formant to the first formant.
TIMING	
length	The length of the utterance in milliseconds.
noSyllsIntensity	The number of syllables in the utterance estimated by counting local maxima in intensity.
speakingRate	The number of syllables per second estimated by counting local maxima in intensity and dividing by the length of the utterance.
startSpeech	The time in milliseconds until speech commenced.
SPEECH PATHOLOGY	
jitter	A measure of periodicity disturbances in the acoustic signal arising from variations of the fundamental frequency.
maximumHarmonicity	The mean harmonics-to-noise ratio in the signal.
meanHarmonicity	The mean harmonics-to-noise ratio in the signal.
minimumHarmonicity	The mean harmonics-to-noise ratio in the signal.
numberOfVoiceBreaks	The number of voice breaks in the voiced part of the utterance.
percentUnvoicedFrames	A measure of the number of frames during speech when voicing is lost.
percentOfVoiceBreaks	The percentage of the voice breaks (measured in time) as a proportion of the spoken part of the utterance.
shimmer	A measure of periodicity disturbances in the acoustic signal arising from variations of the cycle-to-cycle peak intensity.

Table 5.1: The features extracted from sound files.

Technique	Package	Function
Bagging	ipred	bagging
Boosting	gbm	gbm
Naive Bayes	e1071	naiveBayes
Neural Networks	nnet	nnet
Random Forest	randomForest	randomForest
Support Vector Machines	svm	e1071

Table 5.2: The six machine learning techniques experimented with.

treatment of these issues).

The software package we used, R, has the useful feature that the various contributors of packages endeavor to ensure that functions calls for various machine learning algorithms retain as many common parameters as possible. Typically, one's data will take the form of some large table, which can be used in multiple functions. Example 5.2 shows the function calls to create a logistic regression model using a data set (table) called *training*, and then to carry out classification on another data set called *testing*. Example 5.3 shows exactly the same data being used to create a support vector machine.

(5.2)
```
glm.model <- glm(formula = similarity ~ ., family = binomial, data
= training)
pred <- predict(glm.model, testing)
```

(5.3)
```
svm.model <- svm(formula = similarity ~ ., data = training, kernel
= "sigmoid", gamma = 0.5, cost = 4)
pred <- predict(svm.model, testing)
```

We chose six techniques (see Table 5.2), representing three techniques (naive Bayes, neural networks and support vector machines) very often seen in the literature and adding to them three techniques (bagging, boosting and Random Forest) that represent a leading edge of machine learning work called *ensemble learning*. These latter methods generate many classifiers and then aggregate their results. We briefly review each technique when reporting on the experiments in Section 5.3. All of these techniques could be employed by using R function calls.

5.2 Classification using Logistic Regression

In this section we continue to experiment with logistic regression. First, in Section 5.2.1 we bring together the data from Chapters 3 and 4 and see how well we can classify

```
Call:
glm(formula = similarity ~ meanPitch + maximumPitch + meanF1 +
    meanF3 + meanF5 + ratioF2ToF1 + ratioF3ToF1 + numberOfVoiceBreaks +
    noSyllsIntensity + meanHarmonicity + startSpeech + OOG +
    phonemeSimilarity + OOV, family = binomial, data = set1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.80931  -0.02789   0.26881   0.34411   3.90829

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.0125324  3.6734478  -0.820  0.41217
meanPitch           0.0072619  0.0036423   1.994  0.04618 *
maximumPitch       -0.0041764  0.0026860  -1.555  0.11997
meanF1              0.0064275  0.0040437   1.590  0.11194
meanF3             -0.0021605  0.0012406  -1.742  0.08159 .
meanF5              0.0009063  0.0004460   2.032  0.04217 *
ratioF2ToF1        -0.8060376  0.3768216  -2.139  0.03243 *
ratioF3ToF1         1.4230928  0.8088621   1.759  0.07851 .
numberOfVoiceBreaks -0.2562700  0.0628576  -4.077 4.56e-05 ***
noSyllsIntensity    0.0780452  0.0452979   1.723  0.08490 .
meanHarmonicity    -0.0599226  0.0233768  -2.563  0.01037 *
startSpeech         0.4554928  0.2209602   2.061  0.03926 *
OOG                -2.3966113  0.1413361 -16.957  < 2e-16 ***
phonemeSimilarity   0.7932202  0.2821619   2.811  0.00494 **
OOV                -8.0715867  0.7187005 -11.231  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4939.9  on 3999  degrees of freedom
Residual deviance: 1759.7  on 3985  degrees of freedom
AIC: 1789.7

Number of Fisher Scoring iterations: 8
```

Figure 5.2: The logistic model for Set 1 trained on acoustic and language features.

utterances as likely to be recognized correctly or not. Then in Section 5.2.2 we review the outputs available from Sphinx4 that might help us in the classification task. In Section 5.2.3 we re-run our experiments including the outputs from Sphinx4.

5.2.1 Classification using Logistic Regression over all Features

The first task with this experiment was to collect a data set combining the acoustic features and out-of-language indicators used in Chapters 3 and 4. The full data set covers all the acoustic features as shown in Table 3.1, together with the three metrics from the language domain: a binary out-of-grammar metric, a continuous phoneme distance metric and a binary out-of-vocabulary metric. These are the three metrics experimented with in Chapter 4 and in Figure 5.1 and they are derived respectively from the meta-word recognizer, phoneme recognizer and dlm recognizer. In figure 5.2 they are respectively the last three features: OOG, phonemeSimilarity and OOV. We

worked on two sets of data (Set 1 and Set 2), both being made up of 4000 utterances from adult men and women. These sets afforded us the opportunity to train on one set and test on the other.

In this section we report on the results of our experiments using logistic regression. We use our standard methodology of building a logistic model using all the features and then using Akaike's Information Criterion for feature reduction. We then use the model to classify the other half of the data and report on the results.

Training on Set 1

The model derived from Set 1 and used for classification of Set 2 is shown in Figure 5.2. The significance of each factor left in the model is indicated by the number of asterisks shown after it. Interestingly, the most significant factors include the indicators associated with out-of-grammar and out-of-vocabulary utterances, and one associated with speech pathology (the number of voice breaks). The overall performance of the model was very good with an prediction accuracy of 91.42% and a Kappa of 78.41%. Table 5.4 shows the actual outcomes.

Training on Set 2

The model derived from Set 2 and used for classification of Set 1 is shown in Figure 5.3. The most significant factors continue to include the features associated with out-of-grammar utterances, but now features associated with the general disposition of the speaker's vocal equipment become more significant, and the number of syllables in the utterance is important. The overall performance of the model was very good with a prediction accuracy of 90.45% and a Kappa of 76.23%. Table 5.5 shows the actual outcomes.

Summary

Table 5.3 lists the features that were considered significant in each case in order of significance. As can be seen, the information coming from the language model domain is ranked at the top of the features in both cases. Other features, such as speaking rate, pitch and formants, are also present, but appear at differing levels of significance.

5.2.2 The Automatic Speech Recognizer Outputs

Commercial speech recognizers normally produce some form of confidence score for each utterance. Such metrics are based upon a range of outputs from the recognition process and we have dealt with them generally in Section 2.3. Confidence shows a good correlation with the presence of errors and has been used in classification studies by

Training on Set 2	Training on Set 1
OOV	OOV
OOG	OOG
noSyllsIntensity	numberOfVoiceBreaks
meanF1	phonemeSimilarity
meanF4	meanHarmonicity
meanHarmonicity	startSpeech
ratioF3toF1	meanF5
meanPitch	meanPitch
meanF3	rationF3toF1
phonemeSimilarity	meanF3
maximumPitch	ratioF2toF1
meanF5	noSyllsIntensity
	meanF1
	maximumPitch

Table 5.3: Predictive features ranked by significance.

	Actually Incorrect	Actually Correct
Predicted Incorrect	908	16
Predicted Correct	327	2749

Table 5.4: Using logistic regression to predict errors when trained on Set 1.

many researchers. Inclusion appeared to have improved classification in other studies, so we decided to include such a metric in our final classification experiments.

Sphinx4 is an open source recognizer created to further research and was not designed to produce confidence metrics,[1] but it does generate scores associated with each stage in the generation of a recognition hypothesis. These scores are used as the basis for ranking hypotheses, with the best scored hypothesis being offered as the recognition. Sphinx4 computes two such scores:

The acoustic score. The acoustic score represents how likely it is that the speech for the given state matches the acoustic models associated with the given unit of speech represented by that state.

[1]Sphinx4 can be operated using many different configurations and certain of these configurations have had a confidence metric incorporated into them. However, they have only been used experimentally.

```
Call:
glm(formula = similarity ~ meanPitch + maximumPitch + meanF1 +
    meanF3 + meanF4 + meanF5 + ratioF3ToF1 + noSyllsIntensity +
    meanHarmonicity + OOG + phonemeSimilarity + OOV, family = binomial,
    data = set2)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.90234  -0.02737   0.25419   0.33922   3.90852

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.7616071  3.6306498  -0.761 0.446874
meanPitch          0.0100420  0.0038306   2.622 0.008754 **
maximumPitch      -0.0050388  0.0027449  -1.836 0.066403 .
meanF1             0.0139483  0.0039577   3.524 0.000425 ***
meanF3            -0.0031996  0.0012456  -2.569 0.010206 *
meanF4            -0.0020911  0.0007549  -2.770 0.005606 **
meanF5             0.0009996  0.0004838   2.066 0.038821 *
ratioF3ToF1        2.2326552  0.7695360   2.901 0.003716 **
noSyllsIntensity  -0.1201618  0.0284266  -4.227 2.37e-05 ***
meanHarmonicity   -0.0598392  0.0221573  -2.701 0.006920 **
OOG               -2.5707066  0.1450411 -17.724  < 2e-16 ***
phonemeSimilarity  0.6820790  0.2683338   2.542 0.011025 *
OOV               -8.0745303  0.7159713 -11.278  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4944.8  on 3999  degrees of freedom
Residual deviance: 1716.9  on 3987  degrees of freedom
AIC: 1742.9

Number of Fisher Scoring iterations: 8
```

Figure 5.3: The logistic model for Set 2 trained on acoustic and language features.

The language model score. All n-gram language models have probabilities associated with each word and they are used in ranking the competing hypotheses. Sphinx4 does allow one to add probabilities in grammar-based systems, but in practice, this is not often done.

In addition, Sphinx4 also uses an insertion probability. One of the difficulties in recognition is to decide when a word has ended. Sphinx4 has a parameter called the insertion probability that allows one to tune this. A high probability might end up with an utterance being recognized as *four tea cups*, a low one as *forty cups*.

The grammars we used had no probabilities associated with the words they contained, so there was no language model score to use, but we wished to see if the outputs from the speech recognizer, which we had not used until this point, could classify errors. In this experiment this amounted to seeing how well the acoustic scores achieved this. We were trying to match these metrics across utterances of different lengths so we calculate average acoustic scores across all the utterances.

	Actually Incorrect	Actually Correct
Predicted Incorrect	913	63
Predicted Correct	319	2705

Table 5.5: Using logistic regression to predict errors when trained on Set 2.

	Acoustic Features	Acoustic Score
Accuracy	73.13%	69.97%
Kappa	28.72%	7.50%

Table 5.6: Comparing the prediction accuracy of the automatic speech recognizer's acoustic score with the set of acoustic features, training on Set 1 testing on Set 2.

We used these average acoustic scores as one of the features in the data set used in the rest of this chapter and will report on the effect of doing so in due course. However, one question that remained is whether or not the acoustic score was covering the same ground as our acoustic features and vice versa. Logically, one might expect some duplication as both are exploring the space occupied by the acoustic nature of the utterance. We ran an experiment where we first used our set of acoustic features to build a logistic model on one half of our data and then classified the other half; then we did the same using the acoustic scores from the automatic speech recognizer. We then repeated the experiment using the second half of the data to train and the first to test. The results are shown in Tables 5.6 and 5.7. Given the significant drop in Kappa when using the automatic speech recognizer's acoustic score, it was clear that our set of acoustic features is providing more information. Even when we tried to maximize the acoustic score results for Kappa, we only achieved a Kappa a little over 22% with prediction accuracy reducing to just above 61%.

5.2.3 Classification when Including the Speech Recognizer's Outputs

Next we re-ran the experiment described in Section 5.2.1, but now the full data set covers all the acoustic features used in Chapter 3, together with the acoustic score from the automatic speech recognizer and the three metrics from the language domain used in Chapter 4: a binary out-of-grammar metric, a continuous phoneme distance metric and a binary out-of-vocabulary metric. We worked on the same two sets of data (Set 1 and Set 2), both being made up of 4000 utterances from adult men and women. We used our standard methodology of building a logistic model using all the features and

	Acoustic Features	Acoustic Score
Accuracy	73.90%	69.48%
Kappa	30.11%	2.6%

Table 5.7: Comparing the prediction accuracy of the automatic speech recognizer's acoustic score with the set of acoustic features, training on Set 2 testing on Set 1.

then using Akaike's Information Criterion for feature reduction. We then ran the model to classify the other half of the data.

Training on Set 1

The model derived from Set 1 and used for classification of Set 2 is shown in Figure 5.4. It is interesting to note that it does not include the acoustic score from the automatic speech recognizer. The significance of each factor left in the model is indicated by the stars shown after it. As before, the most significant factors include our indicators associated with out-of-grammar utterances and one associated with speech pathology (the number of voice breaks). The acoustic score has been removed from the model by AIC. Given that we determined above that the set of acoustic features used in this thesis were more predictive than the acoustic score, this is not surprising. The overall performance of the model was very good with a prediction accuracy of 91.42% and a Kappa of 78.41%. The model is, in fact, exactly the same as that produced previously when the acoustic score was not included in the data. Table 5.8 shows the actual outcomes.

Training on Set 2

The model derived from Set 2 and used for classification of Set 1 is shown in Figure 5.5. It still includes the acoustic score from the automatic speech recognizer, but at a fairly low level of significance. The most significant factors continue to include the features associated with out-of-grammar utterances, but now features associated with the general disposition of the speaker's vocal equipment become more significant. The overall performance of the model was very good with a prediction accuracy of 90.33% and a Kappa of 76.01%, but it should be noted that its performance is now very slightly worse than when the model was built without using the acoustic score at all. Table 5.9 shows the actual outcomes.

```
Call:
glm(formula = similarity ~ meanPitch + maximumPitch + meanF1 +
    meanF3 + meanF5 + ratioF2ToF1 + ratioF3ToF1 + numberOfVoiceBreaks +
    noSyllsIntensity + meanHarmonicity + startSpeech + OOG +
    phonemeSimilarity + OOV, family = binomial, data = set1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.80931  -0.02789   0.26881   0.34411   3.90829

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -3.0125324  3.6734478  -0.820  0.41217
meanPitch            0.0072619  0.0036423   1.994  0.04618 *
maximumPitch        -0.0041764  0.0026860  -1.555  0.11997
meanF1               0.0064275  0.0040437   1.590  0.11194
meanF3              -0.0021605  0.0012406  -1.742  0.08159 .
meanF5               0.0009063  0.0004460   2.032  0.04217 *
ratioF2ToF1         -0.8060376  0.3768216  -2.139  0.03243 *
ratioF3ToF1          1.4230928  0.8088621   1.759  0.07851 .
numberOfVoiceBreaks -0.2562700  0.0628576  -4.077 4.56e-05 ***
noSyllsIntensity     0.0780452  0.0452979   1.723  0.08490 .
meanHarmonicity     -0.0599226  0.0233768  -2.563  0.01037 *
startSpeech          0.4554928  0.2209602   2.061  0.03926 *
OOG                 -2.3966113  0.1413361 -16.957  < 2e-16 ***
phonemeSimilarity    0.7932202  0.2821619   2.811  0.00494 **
OOV                 -8.0715867  0.7187005 -11.231  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4939.9  on 3999  degrees of freedom
Residual deviance: 1759.7  on 3985  degrees of freedom
AIC: 1789.7

Number of Fisher Scoring iterations: 8
```

Figure 5.4: The logistic model for Set 1 trained on acoustic and language features and acoustic distance from the automatic speech recognizer.

5.2.4 Summary

The experiments reported on in this section show that logistic regression is successful in classifying previously unheard utterances as correctly recognized or not with a prediction accuracy averaging 90.8% and Kappa averaging 77.2%.[2] We saw that logistic regression clearly partitions the utterances with considerable prediction accuracy. However, we also saw that the speech recognizer's acoustic score was not making much contribution to improving the prediction accuracy of the logistic model, and may have had a slightly deleterious effect.

Next we turn to using alternate machine learning methods using these results as a benchmark.

[2]These figures are based upon the second round of experiments, including the acoustic score.

	Actually Incorrect	Actually Correct
Predicted Incorrect	908	16
Predicted Correct	327	2749

Table 5.8: Using logistic regression to predict errors when trained on Set 1.

	Actually Incorrect	Actually Correct
Predicted Incorrect	919	74
Predicted Correct	313	2694

Table 5.9: Using logistic regression to predict errors when trained on Set 2.

5.3 Experimenting with Different Machine Learning Techniques

There are a number of relatively well-known machine learning algorithms, including naive Bayes, neural networks and support vector machines, but there has also been work done on *ensemble learning*, that is, methods that generate many classifiers and aggregate their results. Examples of these techniques are boosting and bagging (see, for example, Schapire [1999], Freund and Schapire [1999] and Meir and Rätsch [2003]). In boosting, new trees are trained based upon the classification performance of past trees. In bagging, later trees do not depend on earlier trees, each being constructed by bootstrapping on a sample of the data. In the end, a majority vote is taken. Random Forest, our third ensemble method, is another tree based algorithm creating a large number of trees from random samples of input cases once again with a majority vote used for classification. We included both traditional machine learning algorithms and ensemble learning methods in our experiments. We decided to use the three relatively well-known techniques plus the three ensemble learning techniques.

5.3.1 Bagging

Method. Bagging is an ensemble method; bagging stands for *bootstrap aggregation* [Berk, 2005, page 17]. First proposed by Brieman [1996], it is best understood by considering the algorithm. Consider the following steps in a fitting algorithm with a data set having n observations and a binary response variable:

1. Take a random sample of size n.

2. Construct a classification tree and do not prune it.

```
glm(formula = similarity ~ meanPitch + maximumPitch + meanF1 +
    meanF3 + meanF4 + meanF5 + ratioF3ToF1 + numberOfVoiceBreaks +
    noSyllsIntensity + meanHarmonicity + accousticScore + OOG +
    phonemeSimilarity + OOV, family = binomial, data = set2)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.85712  -0.02732   0.25121   0.33605   3.89516

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.2306764  3.6666214  -0.608 0.542940
meanPitch            0.0100613  0.0038493   2.614 0.008955 **
maximumPitch        -0.0047069  0.0028178  -1.670 0.094831 .
meanF1               0.0132646  0.0039554   3.354 0.000798 ***
meanF3              -0.0029205  0.0012619  -2.314 0.020644 *
meanF4              -0.0020675  0.0007562  -2.734 0.006253 **
meanF5               0.0010257  0.0004869   2.107 0.035144 *
ratioF3ToF1          2.1324051  0.7678709   2.777 0.005486 **
numberOfVoiceBreaks -0.1202488  0.0681819  -1.764 0.077792 .
noSyllsIntensity    -0.1073567  0.0479402  -2.239 0.025131 *
meanHarmonicity     -0.0628872  0.0222260  -2.829 0.004663 **
accousticScore      -0.0002557  0.0001370  -1.866 0.061978 .
OOG                 -2.6152494  0.1473557 -17.748  < 2e-16 ***
phonemeSimilarity    0.6844796  0.2693749   2.541 0.011054 *
OOV                 -8.0521007  0.7169839 -11.231  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4944.8  on 3999  degrees of freedom
Residual deviance: 1711.9  on 3985  degrees of freedom
AIC: 1741.9

Number of Fisher Scoring iterations: 8
```

Figure 5.5: The logistic model for Set 2 trained on acoustic and language features and acoustic distance from the automatic speech recognize.

3. Repeat steps 1–2 a large number of times.

4. For each case in the data set, count the number of times that it is classified in one category and the number of times it is classified in the other category.

5. Assign each case to a category by a majority vote.

An additional operation is often introduced. At each step, observations not included in the bootstrap sample (called *out-of-bag* observations) are 'dropped' down the tree. A record is kept of the class to which each out-of-bag observation is assigned. This information can be used to compute a more appropriate measure of the classification error, because it derives from the data not used to build the tree.

The basic output from bagging is simply the predicted classes for each case. Commonly, there is also an estimate of the classification error and a cross-tabulation of the

	Actually Incorrect	Actually Correct
Predicted Incorrect	972	63
Predicted Correct	263	2702

Table 5.10: Using bagging to predict errors when trained on Set 1.

classes predicted by the classes observed. The cross tabulation can be useful for comparing the number of false positives to the number of false negatives. Sometimes the software stores each of the trees as well, although they are rarely of any interest because the amount of information is typically overwhelming.

Berk [2005] points to the R package ipred as one effective implementation of bagging. This is the package we used.

Experiment. We trained on one set and tested on the other: we obtained accuracies of 91.9% and 91.5% with Kappas of 80.0% and 79.3% when testing on Set 2 and Set 1 respectively. The actual results are shown in Tables 5.10 and 5.11.

5.3.2 Boosting

Method. Boosting is a way of combining many weak classifiers to produce a powerful committee. Friedman et al. [2000] provide us with a brief history commencing with Schapire's first simple boosting procedure. This shows that a weak learner[3] could always improve its performance by training two additional classifiers on filtered versions of the input stream. After learning an initial classifier h_1 on the first N training points,

- h_2 is learned on a new sample of N points, half of which are misclassified by h_1,
- h_3 is learned on N points for which h_1 and h_2 disagree, and
- the boosted classifier is $h_B = \mathit{Majority\ Vote}(h_1, h_2, h_3)$.

[3]A weak learner is an algorithm for producing a two-class classifier with performance guaranteed (with a high probability) to be significantly better than tossing a coin.

	Actually Incorrect	Actually Correct
Predicted Incorrect	974	81
Predicted Correct	258	2687

Table 5.11: Using bagging to predict errors when trained on Set 2.

	Actually Incorrect	Actually Correct
Predicted Incorrect	932	41
Predicted Correct	303	2724

Table 5.12: Using generalized boosted regression to predict errors when trained on Set 1.

	Actually Incorrect	Actually Correct
Predicted Incorrect	951	80
Predicted Correct	281	2688

Table 5.13: Using generalized boosted regression to predict errors when trained on Set 2.

Schapire [1990]'s *Strength of Learnability* theorem proves that h_B has improved performance over h_1.

Freund and Schapire [1996] proposed a *boost by majority* variation that combined many weak learners simultaneously and improved performance. There is theory in the literature (see for example Freund and Schapire [1996] and Breiman [1997]) to support the approach, but Friedman et al. [2000] report that boosting appears to achieve results far more impressive than the theory would imply.

We used a generalized boosted regression model. The R implementation closely follows Friedman's Gradient Boosting Machine [Friedman, 2001]. Once again the method employs decision trees as its basic classification technique.

Experiment. We ran our training on Set 1 and testing on Set 2 and vice versa. Accuracies came in at 91.4% and 90.9% with Kappas of 78.6% and 77.8% respectively. The actual results are shown in Tables 5.12 and 5.13.

5.3.3 Naive Bayes

Method. A Bayesian Network is a directed, acyclic graph that compactly represents a probabilistic distribution. Figure 5.6 shows an example of such a graph. It deals with the chance of it raining today and tomorrow, but could represent any binary outcome with four original determining factors and two intermediate nodes. E is the event (raining); !E is the not-event (not raining).[4] A Bayesian classifier is such a network applied to a classification task.

[4]The illustration is from `http://www.niedermayer.ca/papers/bayesian/bayes.html#fn10`.

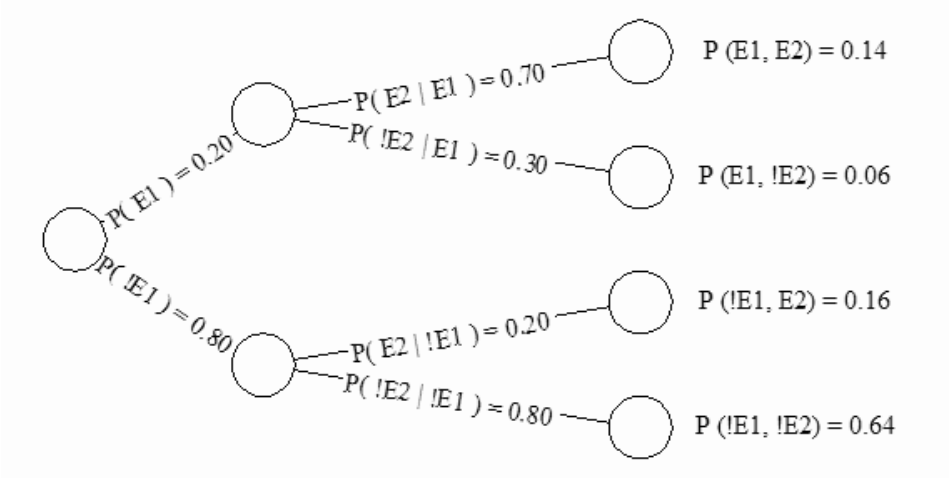

Figure 5.6: An example of a Bayesian Network.

	Actually Incorrect	Actually Correct
Predicted Incorrect	943	236
Predicted Correct	292	2529

Table 5.14: Using naive Bayes to predict errors when trained on Set 1.

A naive Bayes classifier assumes each predictive feature is a direct predecessor of the outcome and also assumes independence amongst features. This very considerably simplifies the model's construction and tractability. Even though these assumptions are generally incorrect, naive Bayes classifiers often work well [Domingoes and Pazzani, 1997; Rish, 2001].

The implementation in R, by David Meyer, is called naiveBayes and is part of the e1071 package. It assumes independence of predictor variables and a Gaussian distribution of any metric predictors.

Experiment. The naive Bayes classifier gave reasonable results. Training on Set 1 and testing on Set 2 achieved 86.8% prediction accuracy and a Kappa of 68.7%. When run the other way around the figures are 86.5% prediction accuracy and 68.0% Kappa. The actual results are shown in Tables 5.14 and 5.15. This is considerably better than the results achieved by Chen and Hasegawa-Johnson [2004] who used a naive Bayes classifier to identify errors. They achieved prediction accuracy of around 75% with a precision of 41%. However, they were working on a much poorer set of features restricted to the outputs of the automatic speech recognizer.

Of course, one of the basic assumptions of naive Bayes is that the predictive factors are independent. We know that is clearly incorrect in our domain. A person's formants

	Actually Incorrect	Actually Correct
Predicted Incorrect	936	244
Predicted Correct	296	2524

Table 5.15: Using naive Bayes to predict errors when trained on Set 2.

all depend upon a common cause, their pitch and vocal tract. However, one may assume that the factors associated with acoustic errors are independent of the factors associated with language errors. Therefore, we looked at creating a Bayesian Network where the classifications for these two types of errors were treated as intermediary nodes. The approach proved to be quite beneficial. Classification on the first half of the corpus improved to 91.3% prediction accuracy and 77.8% Kappa and on the second half the figures were 91.2% prediction accuracy with a Kappa of 77.7%. The actual results are shown in Tables 5.16 and 5.17. Of course, this still leaves the question of what we might achieve by developing a full Bayesian Network. Clearly, many of the features in the acoustic domain are independent; length is independent from intensity, or pitch and so on. Exploring this remains as future work.

5.3.4 Neural Networks

Method. A neural network is an information processing paradigm inspired by the way biological nervous systems, such as the brain, process information. Such information processing system are composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks learn by example and are configured for specific applications, such as data classification, by training. Learning involves adjustments to the synaptic connections that exist between the neurones.

The family of models that neural networks belong to are called nonlinear statistical models. The basic statistical setup for a random continuous response Y with predictor x is:

	Actually Incorrect	Actually Correct
Predicted Incorrect	886	2
Predicted Correct	349	2763

Table 5.16: Using a simple Bayesian Network to predict errors when trained on Set 1.

	Actually Incorrect	Actually Correct
Predicted Incorrect	885	2
Predicted Correct	347	2766

Table 5.17: Using a simple Bayesian Network to predict errors when trained on Set 2.

$$Y = f(x;\theta) + \varepsilon, f \in \mathcal{F}_\Theta, \theta \in \Theta \tag{5.4}$$

where $\mathcal{F}_\Theta$ is a family of functions indexed by a parameter $\theta \in \Theta$, and ε is a random error term. Key challenges in the application of nonlinear statistical models include specifying models for errors, motivating restrictions to function families $\mathcal{F}_\Theta$, and optimizing the resulting objective functions. There is a complex taxonomy of these models. Figure 5.7 schematizes a "generic feed-forward network" after Figure 5.1 of Ripley [1996]. Let y_k denote the kth output element; then the model corresponding to Figure 5.7 is:

$$y_k = f_k[\alpha_k + \sum_{j \to k} w_{jk} f_j(\alpha_j + \sum_{i \to j} w_{ij} x_i)] \tag{5.5}$$

where i, j and k are the input, hidden and output layers, f has a linear or nonlinear form (the logistic function $f(x) = (1 + e^{-x})^{-1}$ is commonly used), x_i is the ith input feature, $\sum_{j \to k}$ denotes summation restricted to connected units, and α and ω are parameters to be estimated.

The most common type of neural network consists of three groups, or layers, of units: a layer of *input* units (i_1 to i_5 in Figure 5.7) is connected to a layer of *hidden* units (j_1 to j_3 in Figure 5.7), which is connected to a layer of *output* units (k_1 to k_3 in Figure 5.7).

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

	Actually Incorrect	Actually Correct
Predicted Incorrect	926	29
Predicted Correct	309	2736

Table 5.18: Using neural networks to predict errors when trained on Set 1.

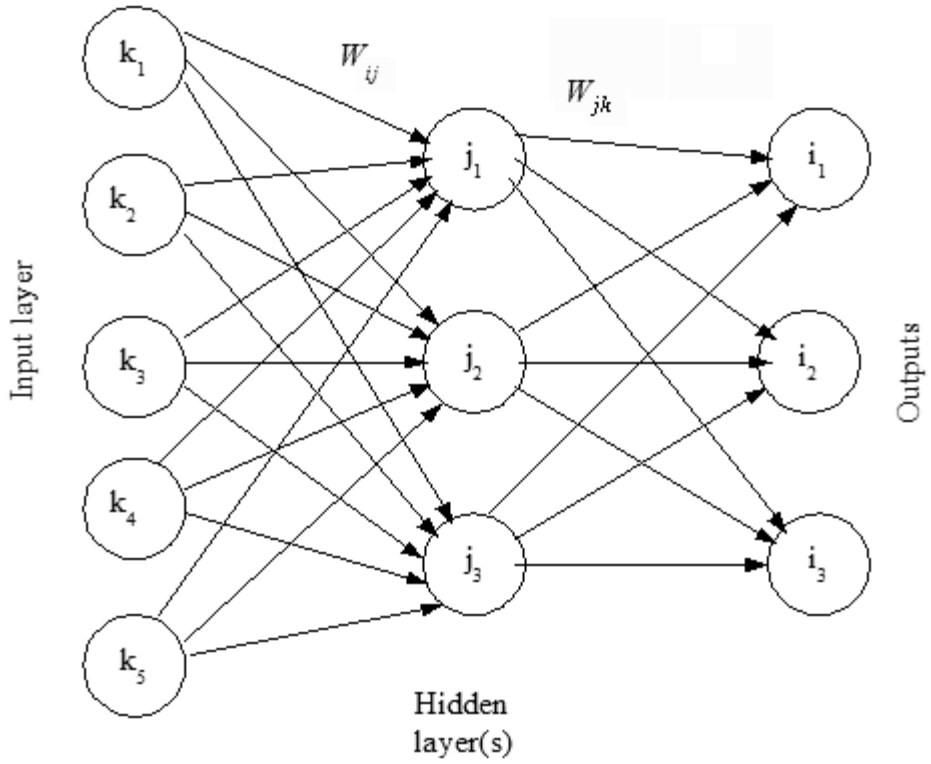

Figure 5.7: Generic feed-forward neural network with one hidden layer of size 3, five inputs and three outputs.

- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

The hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

The R implementation (nnet) is a single-hidden-layer network (see Ripley [1996] and Venables and Ripley [2002]). Training is done using the BFGS method available as part of the R optim function. This is a quasi-Newton method (also known as a variable metric algorithm), published simultaneously in 1970 by Broyden [1970], Fletcher [1970], Goldfarb [1970] and Shanno [1970]. This uses function values and gradients to build up a picture of the surface to be optimized.

Experiment. Once again we trained on one set and tested on the other: we obtained accuracies of 91.6% and 90.5% with Kappas of 78.9% and 77.2% when testing on Set 2 and Set 1 respectively. The actual results are shown in Tables 5.18 and 5.19.

	Actually Incorrect	Actually Correct
Predicted Incorrect	999	149
Predicted Correct	233	2619

Table 5.19: Using neural networks to predict errors when trained on Set 2.

	Actually Incorrect	Actually Correct
Predicted Incorrect	935	17
Predicted Correct	300	2748

Table 5.20: Using Random Forest to predict errors when trained on Set 1.

5.3.5 Random Forest

Method. Tree-structured models have a long history. Breiman et al. [1994] defined a classification and regression tree (CART) procedure. The basic output is a sequence of predicates that define the nodes (splits) and leaves (terminal groupings) of a binary tree. A generalization of the CART procedure is the *Random Forest*[5] methodology of Breiman [2001, 2002]. Whereas CART uses all the variables and all the relevant cases when creating nodes in a single tree that represents the outcome of the learning process, Random Forests creates a large number of trees developed on random samples of the input cases. The input data for each tree is based on a bootstrap sample from the original data. The variables used for constructing splits are a random subsample of the complete set of variables. All trees are grown fully, with no pruning. The classification for a given feature vector is given by the majority vote over all trees.

A decision tree is a tree-like graph that starts at a root; branches run to inner nodes, which in turn allow further branching based upon observation of the association of some predictive variable with an outcome. Leaves represent the predicted value of the outcome, given the values of the variables represented by the path from the root. The parameters for such branching are induced by the algorithm being employed.

Breiman [2001, 2002] proposed the algorithm for Random Forest, which adds an additional layer of randomness to bagging. With standard trees, each node is split using the best split amongst all variables. Here, each is split using a randomly chosen subset of the predictors. This turns out to perform very well and is robust against overfitting. In addition it has few tuning parameters making it very user friendly. The

[5]Random Forest is a registered trade name.

	Actually Incorrect	Actually Correct
Predicted Incorrect	981	102
Predicted Correct	251	2666

Table 5.21: Using Random Forest to predict errors when trained on Set 2.

technique can be more fully reviewed at Breiman's website.[6] Originally Random Forest was implemented in Fortran by Leo Breiman and Adele Cutler; the method has been implemented for R by Andy Liaw and Matthew Wiener. Notes concerning the method can be found in the R documentation.

Experiment. Essentially, we carried out the same experiments here as described previously in this chapter. We trained on Set 1 and tested with Set 2. Random Forest turned out to be fairly demanding of memory and we found we could not hold all the vectors for 4000 utterances in memory.[7] However, we could work with up to 2000 utterances, so we trained on the first half of each set. This memory problem does not occur when classifying.

The approach yielded very good results. When training on half of the first part of the Adult TIDIGITS and testing by predicting the results in the second half of the corpus, we achieved 92.1% prediction accuracy and Kappa of 80.2%. When run the other way around the figures were 91.2% prediction accuracy and 78.6% Kappa, showing very useful consistency. The actual results are shown in Tables 5.20 and 5.21.

5.3.6 Support Vector Machines

Method. This is the family of methods where the function for classification is as follows:

$$g(x) = w^t x + w_0 \tag{5.6}$$

where $x \in \mathcal{R}^p$ is a p-dimensional feature vector, w is a p-dimensional weight vector, and w_0 is called a "threshold weight". For a two-category problem, classification proceeds by determining the sign of $g(x)$. For $K > 2$ categories, category-specific weight vectors and threshold weights are defined leading to the system:

$$g_i(x) = w_i^t x + w_{0i}, i = 1, ..., K \tag{5.7}$$

and classification proceeds by determining the value of i for which g_i is maximized.

[6] `http://stat-www.berkley.edu/users/breiman/RandomForests/cc_home.htm`.

[7] We were working with 512 MB of RAM.

	Actually Incorrect	Actually Correct
Predicted Incorrect	913	11
Predicted Correct	322	2754

Table 5.22: Using support vector machines to predict errors when trained on Set 1.

The classifier definition in Equation 5.6 leads to decision boundaries that are hyperplanes in a (possibly transformed) feature space. Such boundaries may be too rigid in certain applications. Local methods of estimation and classification provide greater flexibility. A basic tool for localization is the concept of a kernel function. A kernel K is a bounded function on the feature space, which integrates to unity (see Section 6.1 of Ripley [1996]). Gaussian density is a frequently encountered kernel. Kernel estimation of probability densities is widely practiced; the *Parzen estimator* of a density on the basis of N points has the form:

$$\hat{f}(x) = (N\lambda)^{-1} \sum_i K_\lambda(x, x_i) \tag{5.8}$$

where λ plays the role of a bandwidth, e.g., $K_\lambda(x, y) = \phi(|x - y|/\lambda)$, where ϕ is the standard Gaussian density. For classification, the class-specific density estimates can be used to compute Bayes Rule. Support vector machines are one of the techniques in this family.

We used the function svm contained in the R package e1071.[8] There is considerable material available on the use of support vector machines, ranging from very detailed treatments like Burges [1998] to more direct guides like Chih-Wei et al. [2003] and even Internet tutorials.[9]

The choice of kernel and the setting of the parameters of the kernel can be vitally important for the success of the approach. There are four kernels used in training and predicting that are available with the e1071 implementation with the following

[8]Developed by David Meyer at the Vienna University of Technology.

[9]See for example *SVM-Tutorial using R (e1071-package)* at `http://www.potschi.de/svmtut/svmtut.html`.

	Actually Incorrect	Actually Correct
Predicted Incorrect	924	5
Predicted Correct	308	2763

Table 5.23: Using support vector machines to predict errors when trained on Set 2.

parameters:

linear $u'v$

polynomial $(\gamma u'v + coef0)^{degree}$

radial basis $e^{(-\gamma|u-v|^2)}$

sigmoid $tanh(\gamma u'v + coef0)$

The linear kernel is simply a special case of the polynomial kernel. Polynomial kernels are computationally expensive and higher order polynomials have often been found not to produce better results. Certainly this was true in our case. From some of our previous unpublished work we expected the radial kernel to perform best. This kernel performs best with Gaussian distributions and we have already seen how assuming such a distribution produced the best results when using boosting.

We tried the four different kernels on their default settings, which further confirmed the choice of kernel. The radial kernel proved most successful and is reported on below. Radial kernels may indicate that we are investigating a highly non-linear structure. Normally we have found support vector machines very sensitive to parameter settings, but in this case we appeared unable to improve performance by varying them.

Experiment. Once again we trained on one set and tested on the other: we obtained accuracies of 91.7% and 92.2% with Kappas of 79.0% and 80.3% when testing on Set 2 and Set 1 respectively. The actual results are shown in Tables 5.22 and 5.23.

5.3.7 Summary

In this section we reported on six machine learning algorithms and the experiments we carried out using them. They were bagging, boosting, naive Bayes, neural networks, Random Forest and support vector machines. Table 5.24 provides a summary of the average accuracies and Kappas for these machine learning techniques.

As can be seen, the performance of all of these methods was very similar, in the low 90 percents (save for naive Bayes in its 'out-of-the-box' form). The most successful was support vector machines with 92.0% prediction accuracy and 79.7% Kappa. The techniques did generally show a small but distinct improvement over logistic regression, which produced an average prediction accuracy of 90.8% with a Kappa of 77.2%.

5.4 Evaluation

While it is clear from the previous sections that we could identify utterances that were likely to be misrecognized, the data we worked on was of a high (recording) quality and the language, being spoken digits, quite restricted. However, we did have the sound

Method	Accuracy	Kappa
Bagging	91.7%	79.7%
Boosting	91.2%	78.2%
Naive Bayes	86.7%	68.4%
Neural Networks	91.1%	78.1%
Random Forest	91.7%	79.4%
Support Vector Machines	92.0%	79.7%

Table 5.24: Relative performance of six machine learning methods on the Adults' TIDIGITS Corpus.

files for the CU Corpus which represented a very real life data set. The CU Corpus came with documented recognitions made using the SONIC speech recognizer. It was originally recognized using an n-gram language model. To subject it to our techniques we required a grammar. If we wrote a grammar, we could recognize the CU Corpus using Sphinx4. Section 5.4.1 reviews the grammar writing task and, as Section 5.4.2 reports, the results of the work using Sphinx4 confirmed our earlier results.

In addition, we wanted to make use of a recognizer's confidence measure so that we could compare our performance directly with a confidence metric used by a commercial recognizer. We had access to a Nuance 8 recognizer and repeated our experiments using it (Section 5.4.3).

Finally, it is notoriously difficult to compare the results of one's experiments with other reported work. It is not only that the literature often reports in a brief fashion: the metrics used may be slightly different, and the corpora almost certainly are. To provide a comparative, we chose to use the classification methods employed in Hirschberg et al. [2004] on our data (Section 5.4.4). Hirschberg et al. report the best classification performance we could find in the literature.

5.4.1 Writing a Grammar for the CU Corpus

Our first task was to associate each utterance with a dialogue state, that is, the sort of utterance that is expected in response to some type of question, such as a question about a place, time or date. We set about hand tagging the sixteen thousand utterances in the corpus. Ultimately we ended up with the seven groups shown in Table 5.25, where the utterance clearly fell into one or another grammar state. There were some 11,831 utterances in these groups. The yesNo state was very much larger than any other grammar state so we decided to concentrate on it. Such a grammar state may on first reaction be considered to be trivial, but it did contain 557 unique utterances and as can

State	Example	Count
airline	frontier airline would be okay	365
car	a budget fullsize car	224
date	august eleventh two thousand	1180
hotel	do they have a marriott	252
place	i'd like to leave seattle and go to london	2099
time	after six o'clock in the evening	1148
yesNo	no	6563

Table 5.25: Grammar states for the CU Corpus.

be seen in Figure 5.9 required a complex grammar to achieve full coverage.

In writing our grammar, we did not attempt to handle disfluencies as shown in Example 5.9, or compound utterances as shown in Example 5.10. The former are typically not covered in commercial grammars and the latter are often avoided in the 'form filling' exercises typical of commercial systems.

(5.9) i wanna leave miami i wanna land in miami

(5.10) i wanna go from washington d c to tel aviv leaving on the twenty third of september

We immediately ran into what we refer to as *yes no confusion*. *Yes*, *yea*, *yep*, *yeah* and *yup* are very phonologically similar, as are *no*, *nope*, and *none*. The differences are of no semantic significance, but under our definition of error[10] they are different.

In any commercial grammar all the variations on *yes* or *no* will lead to the correct semantic outcomes, so we decided to carry out a study of the problem with our yesNo utterances. Taking our strict definition of error, we found 4019 errors in our 6563 yesNo utterances; an utterance error rate of 61.2%. However, the simple expedient of treating *yes* as equivalent to the other phonologically similar affirmative utterances (*yep* and *yeah*), and *no* equivalent to *nope*, reduces the errors to 1168; a 17.7% utterance error rate. This still leaves outstanding semantically similar utterances with no apparent phonological similarity. Example 5.11 shows such a case:

(5.11) utterance: yes please
recognition: that's perfect

[10]We generally discovered errors by using a strict string comparison under which, for example *four* would be considered a misrecognition of the utterance *for*.

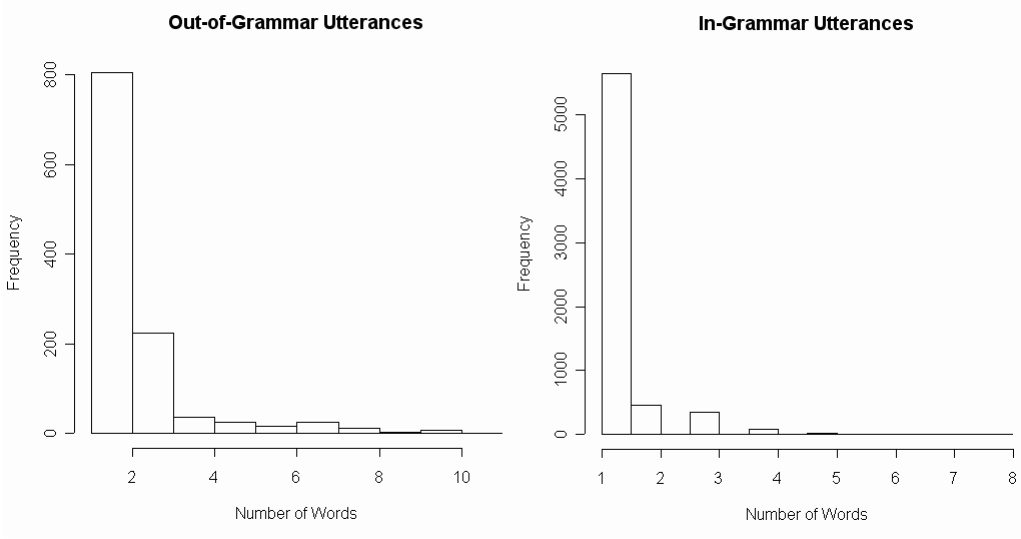

Figure 5.8: Histograms of the length of utterances in the appraisal corpus.

A commercial grammar will return the same semantic value *yes* for either the utterance or the recognition in Example 5.11, and if we were to take the semantic values likely to be returned by a commercial grammar across our yesNo utterances the number of errors drops to 553; an error rate of 8.4%. This clearly demonstrates how one can have apparently high error rates yet still achieve successful dialogues, but left us wondering upon what basis we should characterize errors for our test. In the end, we decided that if the simple expediency of treating all phonologically similar *yes* words as one class and all phonologically similar *no* words as one class reduces errors from 61.2% to 17.7%, any sensible system would adopt it. In consequence, we decided to apply our studies to the yesNo grammar state defining errors so as not to include those that arise simply from phonological differences across the *yes* or *no* answers, but to leave any other differences between a recognition hypothesis and an annotation as an error.

5.4.2 Classifying with the CU Corpus

We took the 6,500 yesNo utterances and, in order to seed them with out-of-grammar utterances, added a little over 1150 utterances from other parts of the corpus, covering responses to questions about airlines (365), car hire (224) and dates (561). These out-of-grammar utterances were chosen at random, but no tests were carried out to see if the average intensity, pitch and so on were similar across the various groups. The average length of the out-of-language utterances was 2.31 words against the balance of the corpus at 1.23. However, as Figure 5.8 shows, the vast majority of utterances in both corpora were one or two words in length. Although our machine learning algorithms might use length as a discriminating factor between in-language and out-of-language

utterances, these counts represented 69.4% and 93.2% of the utterances in these groups respectively, very considerably reducing the value of length as a determinant between the two groups. In addition, anecdotal evidence implicates longer utterances as a source of many out-of-language utterances as users lose their way in a dialogue.

The out-of-grammar utterances give an out-of-grammar component in the corpus of some 15%. We ran the recognizer on the corpus using the grammar shown in Figure 5.9.[11] We used the same phoneme language model as we had previously, but created a new domain language model from the actual corpus for the n-gram recognitions.

We sorted the data by dialogue numbers, so that different speakers would appear sequentially through the data, and decided to train on the first 6,500 utterances and test on the final 1,000, so that we were testing on new speakers. This corpus consisted of speech over mobile and land based telephones; this is probably the most difficult recognition task short of children's speech and, of course, a task often encountered in spoken language systems.

We ran our six machine learning techniques and produced the results shown in Table 5.26. As can be seen, we achieved a best result of 92.2% prediction accuracy with a Kappa of 78.7%.

5.4.3 Classifying with Nuance 8

Nuance 8 is a well-known commercial recognizer used in many commercial applications. It is produced by Nuance, the leading provider of speech solutions for businesses, who made it available to Macquarie University for research purposes. It produces a confidence score as a standard output. When used in grammar mode it uses a grammar formalism called the Nuance Grammar Specification Language, which is very similar to JSGF although there are minor syntactic differences.

Essentially we repeated the experiment reported on in the previous section but now used Nuance 8 as our grammar-based recognizer rather than Sphinx4. It is interesting to note that our utterance error rate on the in-grammar part of our corpus was reduced to 11.3% when using Nuance 8 compared with 17.7% with Sphinx4, in both cases treating phonologically similar words such as *yes* and *yep* as identical.

[11] The formalism in this grammar is based upon each variable (for example *public <deny>)* specifying other variables (for example *<absoluteNo>*) or words allowed in the grammar (for example *(just perfect)*). Words or variable are allowed in the order written, but the use of a pipe (|) makes the words alternatives in the grammar. Variables expand to the words they cover for the purpose of recognition.

```
#JSGF V.10;

grammar yesNo;

public <yesNo> = (<affirm> | <deny>) ;

public <affirm> = (<absoluteYes> | <correct> | <otherYes> | <iYes> | <illYes> | <thatYes>
| <yes> | <sure> | <yeah> | <you>);

public <deny> = (<absoluteNo> | <otherNo> | <forget> | <no> | <iNo> | <illNo> | <thatNo> |
<nope>];

public <absoluteYes> = absolutely;

public <absoluteNo> = absolutely not;

public <correct> = correct [okay];

public <otherYes> = (fine | it's okay | just perfect | maybe | that right | okay |
perfect);

public <otherNo> = incorrect;

public <forget> = forget (about it | that part);

public <no> = [hell] no+ [i am not | i did not | i didn't | i have not | preference |
thank you | thanks | that is not correct | that is not okay | that's not at all correct |
that's not correct | preference | thank you | that is not correct | that is not okay |
that's not right | too expensive];

public <iYes> = i (did | did yes | don't care | sure did thank you | sure would

public <iNo> = i (don't | don't have any | don't know | don't think so | think i can do
that yes | think so

public <illYes> = i'll (take it | think so);

public <illNo> = i'll pass;

public <thatYes> = (that is | that's) (right | perfect | okay | it |great | good | fine |
correct | definitely correct | was correct);

public <thatNo> = that is (not correct | not okay| is wrong);

public <yes> = [finally] yes+ [this is okay | this is correct | that's right thank you |
that's perfect | that's right |that's okay | that's fine | that's correct | that's alright
| that'll be great | that would be nice | that would be great | that will be fine |that
that's fine | that is okay | that is correct | right | please do | please | perfect | yes
please | okay | mamm | it's right | it's okay | it's correct | it would be good | it is
yoo-hoo | it is sure fine | it is right |it is okay | it is correct | it is | i'll take it
| i'd like to | i would* | i will be | i will | i did | it is | i am ] [thank you |
please];

public <sure> = sure (that's correct | thing | why not | yes);

public <nope> = (none | nope) [of them | it is not correct | no];

public <yeah> = (yeah | yep | yes) [definitely | oks | than you | that's cool | that's
correct | that's fine | that's fine i'll take that | that's good | that's ok | that's okay
| that's right |  this is correct | yes];

public <you> = (you | you've) [bet | bet baby | got it | got that right ];
```

Figure 5.9: The grammar for recognizing yesNo utterances in the CU Corpus.

Method	Accuracy	Kappa
Bagging	87.5%	64.1%
Boosting	92.2%	78.7%
Neural Networks	86.6%	60.6%
Naive Bayes	81.5%	38.6%
Random Forest	86.2%	59.1%
Support Vector Machines	86.5%	60.1%

Table 5.26: Relative performance of six machine learning methods on the CU Corpus (yesNo grammar states).

Nuance 8 afforded us the opportunity to use a commercial confidence metric in our predictors. One of our objectives was to be able to compare our method with this metric. We used the concept of breakpoints and false positives, introduced in Section 2.3, to convert the metric into a binary predictor. We calculated at what point confidence performs best as a binary classifier of errors over the first 6,500 utterances. Confidence performed better in this case than in our other experience,[12] and the best prediction accuracy level was at around 52% confidence where we achieved a 92.6% prediction accuracy. Figure 5.10 shows the performance over the full range. We then calculated what prediction accuracy the 52% confidence level produced in a test set consisting of the next 1,000 utterances. It produced a prediction accuracy of 90.8%.

Although we did not time the exercise, as execution time was not of great importance, Nuance ran noticeably faster than Sphinx and we would assume it had been optimized for speed when using grammar-based recognition. In addition, we found that we had a small number of cases where no recognition was offered by the recognizer. All of these cases had a zero confidence. We knew from inspection of these cases that 44 came from our out-of-grammar set and 48 were actually in-grammar. This meant that in 92 cases out of 7721 (1.19% of the total number of utterances) we had no recognition to work with. It may well be that the optimization that had been undertaken resulted in a severe pruning of the search space, meaning in this small number of cases there were simply no hypotheses left to offer as recognitions.

We achieved extremely high levels of prediction accuracy in this experiment: neural networks performed best with 95.4% prediction accuracy and 85.9% Kappa.

[12]Nuance 8 represents a well developed, fielded commercial recognizer. Its method of calculating confidence is not publicized. However, it will have been optimized in many fielded applications.

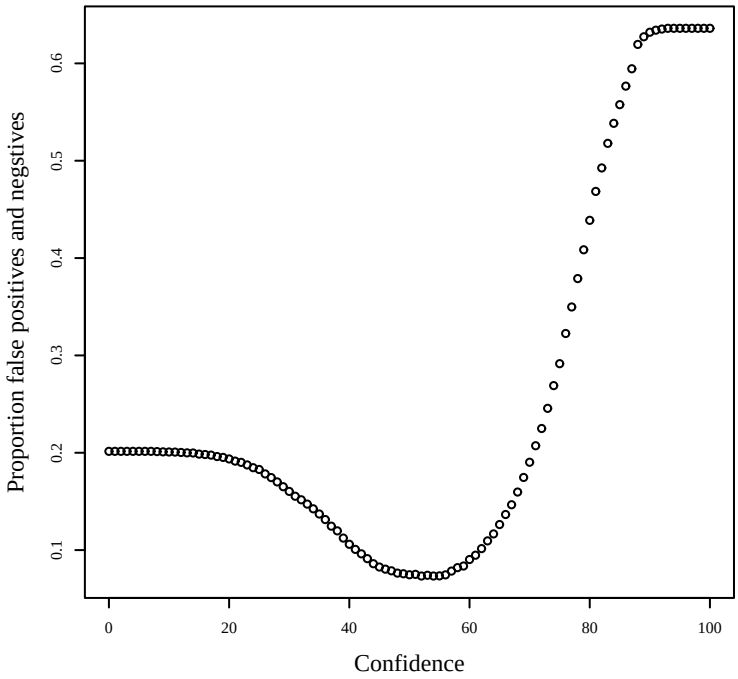

Figure 5.10: The proportion of false results in the first 6,500 utterances at various confidence levels when using a Nuance 8 Recognizer.

5.4.4 Comparison with Hirschberg's Approach

Hirschberg and her associates have published by far the most extensively on prediction of errors from features extracted from acoustic files. Hirschberg et al. [2004] report on two experiments with RIPPER [Cohen, 1996], one on the W99 Corpus[13] and the TOOTS Corpus (a train enquiry service). In the former, they achieved prediction accuracy of 77.2% in classifying utterances as likely to be recognized correctly or not with their best set of features, but they did get up to 91.4% with the latter corpus. This was the highest level of performance we found in the literature.

We wanted to see how we might compare with Hirschberg et al. It is very difficult to compare the outcomes of one researcher's work with another, and while RIPPER, the machine learning software they used, is no longer available, the algorithm has been re-implemented in the WEKA software package[14] under the name JRip. We ran it on the set of features used by Hirschberg et al.:

- maximum and mean fundamental frequency values,

[13] W99 was a spoken dialogue system used to support registration and information access for the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU99).

[14] `http://www.cs.waikato.ac.nz/ml/weka/`.

Method	Accuracy	Kappa
Bagging	95.1%	84.9%
Boosting	91.0%	74.9%
Naive Bayes	92.7%	78.7%
Neural Networks	95.4%	85.9%
Random Forest	94.4%	82.8%
Support Vector Machines	95.2%	85.0%

Table 5.27: Relative performance of six machine learning methods on the CU Corpus (yes/no grammar states) when using a Nuance 8 Recognizer

- maximum and mean energy values,
- total duration,
- length of pause preceding turn,
- speaking rate, calculated in syllables per second,
- amount of silence in the turn,
- grammar state,
- string recognized,
- confidence,
- presence of *yes*, *no*, *cancel* or *help* in the recognition, and
- number of words and syllables in the recognition.

This was the highest performing combination with an prediction accuracy rate of 91.3% on the TOOTS Corpus.

We applied JRip to the Hirschberg et al. feature set which we derived from the data available from the Nuance evaluation. It produced an accuracy of 92.28% with a Kappa of 77.85%. This compares with the performance we reported on in Section 5.4.3 using our feature set where a neural network produced an accuracy of 95.4% and a Kappa of 85.0%. We concluded this set of experiments by carrying out two further experiments to see if we could establish if it was the feature set or the classification method that made the difference in performance. We ran our feature set using JRip and this resulted in a very marginal reduction in accuracy to 92.26% but an equally marginal increase in Kappa to 77.95%. This indicated that a RIPPER-type decision tree algorithm was unable to use the additional information contained in our larger feature set. Finally, we

then ran a neural network using the Hirschberg et al. feature set and it produced an accuracy of 92.6% with a Kappa of 80.36%. We concluded that it was the wider feature set run using a more suitable machine learning algorithm that caused the improved performance of our classifier over the work reported on by Hirschberg et al.

5.5 Outcomes

In this chapter we reported on a series of experiments designed to explore how accurately we could classify utterances as likely to be correctly recognized or not. In Section 5.1, we introduced the chapter, explained our methodology and discussed the classification task.

Initially, we worked on some 4000 adult TIDIGITS utterances containing both acoustic and language errors, training on the first half and testing on the second and vice versa. In Section 5.2 we reported on our experiments using logistic regression, first combining the acoustic and language features previously explored separately in Chapters 3 and 4 respectively. Then we added the automatic speech recognizer's acoustic score to the predictive features. The presence of this additional information appeared to make very little difference, with average prediction accuracy actually reducing from 90.9% to 90.8% and average Kappa unchanged at 77.3%. Sphinx4 only provided us with a simple acoustic score that may even have introduced a little confusion into the model.

We then extended the work by experimenting with six machine learning techniques: bagging, boosting, naive Bayes, neural networks, Random Forest and support vector machines. Section 5.3 explained the six machine learning methods we used and the results of the experiments. Table 5.24 shows the overall results with support vector machines producing the best average prediction accuracy at 92.0% and average Kappa of 79.7%.

Given that our TIDIGITS corpus was studio quality recording of spoken digits, we decided to evaluate the techniques on a more realistic corpus. We took 6,500 utterances from the CU Corpus (telephone speech in the air travel information domain) and ran evaluations (Section 5.4). We ran these first using Sphinx4, and then using a commercial recognizer, Nuance 8. Nuance 8 turned out to be a more accurate recognizer and the best performance achieved, using neural networks, showed an average prediction accuracy of 95.4% and an average Kappa of 85.9%. This compared with a prediction accuracy of 90.8% using confidence alone and 92.2% when using the techniques employed by leading researchers in this field [Hirschberg et al., 2004].

The evaluation with the commercial recognizer indicates that high levels of binary classification can be achieved using the techniques developed in this thesis.

Chapter 6

Conclusions

6.1 Introduction

In this thesis we reported on our work done to produce a highly accurate classifier of recognitions produced by automatic speech recognizers as correct or containing errors. Modern recognizers split the recognition problem into two parts, one to do with matching sounds (principally phonemes) with acoustic models and the other to do with matching the phonemes that make up words with a language model. Throughout, it has been our approach that a technique could be developed by first splitting the problem into two corresponding parts.

In the acoustic domain we identified a corpus of studio recorded utterances (TIDIGITS) that comprised of a limited vocabulary (digits). This allowed us to ensure our language model provided full coverage so we could explore the acoustic causes of misrecognitions. We used logistic regression to analyze and predict.

In the language domain we could then remove any utterances that had been misrecognized for acoustic reasons and, by removing a single digit from our language model, explore and compare several techniques able to automatic discover language errors.

Finally, we brought together what we had learned in the acoustic and language domains and carried out a series of experiments to see how accurately one could classify utterances as likely to be misrecognized or not. Ultimately, using a dialogue state where the user confirms the system's understanding as an example, we achieved prediction accuracy of over 95% in this task.

In the remaining sections of this chapter, in Section 6.2 we review the outcomes of this thesis. In Section 6.3 we review the place of this work in the literature. In Section 6.4 we review a number of the strands of future research that arise from this thesis and in Section 6.5 we provide some final words.

6.2 Outcomes

6.2.1 Starting Points

In Chapter 2 we reviewed the existing related work upon which we hoped to build to produce our classifier. First, we placed our chosen area of work, early stage detection, in context amongst the work on handling errors. This larger work may be characterized as consisting of:

Prevention: getting people to say things the recognizer will find easy to handle.

Detection: this includes early stage detection, the subject of this thesis, and late detection, which concentrates on identifying errors that have already entered the dialogue.

Recovery: this covers creating effective sub-dialogues or automatic methods to repair errors.

Then we looked at work relating to confidence, the current metric generally used to manage errors. We looked at how dialogue designers use its level to trigger different confirmation techniques; how it is generally calculated from data arising solely within the speech recognizer such as acoustic scores calculated during recognition; and how, while it correlates well with errors, it turns out to be a poor classifier. The literature indicates that it often performs with little better than 75% prediction accuracy.

Then we turned to work done in our two principal areas of interest: errors arising in the acoustic domain and errors arising in the language domain. We reviewed the considerable work associated with using acoustic features to classify utterances as likely to be misrecognized. Most of this work is based upon the theory that prosodic variation, particularly that associated with hyperarticulation, is a root cause of such errors. There is less work associated with identifying language errors, where we reviewed the use of sub-word language modeling and large vocabulary recognizers. Finally, we identified early stage recognition of speech recognition errors as significant problem which the literature indicated might hold the promise of being solved.

6.2.2 Errors in the Acoustic Domain

The next three chapters report on the substantive work of this thesis: errors in the acoustic domain, errors in the language domain and a series of experiments in classification. In Chapter 3 we described the four hypotheses we tested through four experiments and their results:

The Consistency Experiment, where by a ten-fold verification, we tested if the features associated with predicting errors were consistent throughout our data: mean

prediction accuracy was 80.2% with a standard deviation of 1.0% and mean Kappa 35.3% with a standard deviation of 2.4%.

The Prediction Experiment, where by training on one half of our data from one group of speakers and testing on the other half comprised of different speakers and vice versa, we tested if the utterances of previously unheard speakers could be classified: mean prediction accuracy was 85.2% with a standard deviation of 0.2% and Kappa of 17.3% with a standard deviation of 0.2%.

The Error-Prone Speakers Experiment, where by looking at the proportion of errors produced by each speaker, we tested if acoustic errors were associated with the general way individual speakers render utterances, rather than the different ways an individual speaker might render utterances at different points in a dialogue: a Pearson Chi Square Test dismissed the hypothesis that errors were equally spread amongst speakers.

The Goat Experiment, where by using a logistic model we tested if we could predict the individual speakers who would experience difficulties being recognized: a regression line accounted for 65% of the variation between predicted and actual speakers.

We first carried out the experiments on a subset of the TIDIGITS Corpus where we ensured our grammar covered all the utterances present; the figures above are from these experiments. This allowed us to work solely in the acoustic domain. Then we repeated the experiments on a subset of the CU Corpus to see if our findings held when facing a more real life data set, including language errors. They did, and these results were also reported in Chapter 3, confirming that:

- the features that reveal errors were consistent throughout this corpus;
- logistic models trained on one group of speakers could predict errors when classifying the speech of different groups of speakers;
- errors were related to some general qualities of the speaker and not of the utterance; and,
- a logistic model could predict which individual speakers would be misrecognized.

Finally in Chapter 3, we reported our findings on the causes of acoustic errors where we established that individual speaker characteristics were a major factor and the significant predictive factors connected with errors were:

- intensity of the sound signal,

- vocal equipment size, and
- speech pathology.

This differs from the view prevalent in the literature that prosodic variation in the nature of hyperarticulation is a major cause.

6.2.3 Errors in the Language Domain

In Chapter 4 we described how we pursued the hypothesis that a second recognition using a more general language model would reveal if the initial recognition had encountered an out-of-language utterance. We reported on the three methods we explored, their results and relative merits:

A meta-word, where a 'word' designed to match any pattern of phonemes is placed into the language model and given a small probability of being encountered. If the recognizer proposes the meta-word, one concludes that an out-of-language utterance has been encountered. We found this method to be 82.3% accurate with a Kappa of 51.7%. This method carries very little overhead in terms of the creation of the meta-word and is domain independent.

A phoneme language model, where a language model is created that allows up to tri-grams of phonemes as found in a large corpus of English. A distance metric is then calculated from the phonemes in the initial recognition and some break-point established below which one concludes that an out-of-language utterance has been encountered. We found this method to be 53.7% accurate with a Kappa of 19.7%. This method carries some overhead in terms of the creation of the language model and the additional calculation required to classify utterances, but is domain independent.

A domain language model, where a language model is created from actual experience in the domain of the application. If a recognition using this shows words that are not in the initial recognition's language model one concludes that an out-of-language utterance has been encountered. We found this method to be 99.5% accurate. This method carries a very considerable overhead in terms of the creation of the language model, as annotation of a large corpus of sound files collected in the domain is essential and it is very domain dependent.

The use of a domain language model is highly dependent upon the rate at which new words are encountered by a spoken dialogue system. We found the rate fell to well under 5% once some 5,000 or 10,000 utterances had been encountered. Finally, we reported on some findings we made in regard to modeling the rate of introduction of new words.

Given that the rate of introduction of such words can be modeled as a negative power relationship, integration can be used to determine such variables as the size of training corpus required to achieve a particular level of coverage.

The work reported on in Chapter 4 provides a comprehensive comparison of the techniques available for identifying out-of-language utterances.

6.2.4 Experiments in Classification

In Chapter 5 we describe a series of experiments in classification. First we combined the features we had used in Chapters 3 and 4 and used logistic regression to analyze and predict. We worked on some 4,000 utterances from adult men and women from the TIDIGITS Corpus, using one half to train on and the other to test on and then vice versa. The corpus included utterances that would be misrecognized because of acoustic distance; in order to ensure a proportion of out-of-language utterances we removed the word *three* from our grammar. We achieved average prediction accuracy in our predictions of 90.9% with a Kappa of 77.4%

We wanted to use all the information to hand to accurately classify utterances as likely to be recognized with or without error. So, in addition to the data we had obtained concerning the acoustic features of utterances and the data concerning out-of-language utterances, we decided to include the acoustic distance metric produced by the speech recognizer. We then re-ran our experiments. The additional information may actually have confused the model as average prediction accuracy reduced fractionally to 90.8% with a Kappa of 77.3

We then moved on to experiment on the same data using six different machine learning methods: bagging, boosting, naive Bayes, neural networks, Random Forest and support vector machines. Naive Bayes performed poorly with prediction accuracy of 86.7%, but the other methods performed at just under 92% with support vector machines achieving 92% with a Kappa of 79.7%.

We then subjected the methods to evaluation using a more real life data set. We took 6,500 utterances from turns in the CU Corpus where the user was answering a direct request for confirmation. We wrote a grammar to cover all of the examples. We then seeded the corpus with a little over 1,150 utterances from the CU Corpus that were out-of-language. We then ran recognitions using Sphinx4, extracted the acoustic features and built our models. We trained on the first 6,500 utterances in our corpus and tested on the last 1,000. This time the best result was from boosting with 92.2% prediction accuracy and a Kappa of 78.7%.

We had never had a useable confidence figure from our recognizers, so we decided to re-run the CU experiment using a commercial recognizer: Nuance 8. Including confidence as a predictive factor we achieved prediction accuracy of 95.4% with a Kappa of 85.0% using neural networks. In addition we reimplemented the techniques used in

Hirschberg et al. [2004] using our own data so that we could compare with this leading research in this field. We achieved a prediction accuracy of 95.4% against 92.2% when using Hirschberg et al.. We report on this more fully in the next section.

6.2.5 Summary

In this thesis we have concentrated on one significant problem encountered with spoken dialogue systems: the occurrence of errors in the recognition being offered by the speech engine. We focused on improving the ability of the system to know that it has misheard. We did this by dividing the problem into two parts, acoustic and language, and carrying out a series of experiments:

1. We carried out a first series of experiments to see how far we could predict error prone utterances solely from the acoustic data in their sound files.

2. We carried a second series of experiments to see how well we could identify the presence of out-of-language utterances by comparison of a tightly focused grammar based recognition and recognitions using other more general language models.

3. Finally we carried out a series of experiments using both acoustic and language data to identify utterances that would be misrecognized.

Our final evaluations achieved accuracies in identifying such utterances of 95.4%.

6.3 The Place of this Work in the Literature

6.3.1 Improvement over other Techniques

We saw in Chapter 2 that the current state of the art technique in detecting 'mishearings' in spoken dialogues with computers is the use of confidence. It would appear from the literature that whatever threshold is chosen, false positives above the threshold and false negatives below it leave this metric often only some 75% accurate. Even on those rare occasions when it reaches a 90% level, one in ten utterances would be wrongly classified as correct when they are wrong or vice versa. Of course as one progressively improves classification the increment will appear smaller, but it should never be forgotten that an increase, for example, from 90% to 95%, halves the problem. A major claim of the present work is to have lifted the levels of prediction accuracy that a binary confidence metric can produce to over 95%. The literature reports on the work others have done to move in this direction.

Hirschberg and her associates achieved prediction accuracy of 77.2% in classifying utterances as likely to be recognized correctly or not with their best set of features on one corpus and 91.3% on another. Given this was one of the best levels of prediction

accuracy achieved we used the same methods to evaluate our performance in Section 5.4.4. There we reported that using Hirschberg et al.'s techniques on our data produced an prediction accuracy of 92.2% with a Kappa of 77.5%. Zhou et al. [2005] used a naive Bayes classifier to identify errors. They achieved prediction accuracy of around 75% with a precision of 41%. However, they worked on a much more limited set of features as they did not use any data from the sound files directly, but were restricted to the outputs of the automatic speech recognizer. Even though the acoustic and language domains are the two dimensions through which the recognition problem has been solved, researchers tend to concentrate on the one or the other domain when looking into the problem of detecting errors (see for example, Krahmer et al. [2001] who worked exclusively in the language domain, or Hirschberg et al. [2004] who worked principally in the acoustic domain with limited language information included). Indeed, an achievement of this thesis is its willingness to tackle both these sources of the cause of errors in automatic speech recognition.

6.3.2 The Causes of Acoustic Errors

In relation to acoustic errors we have given an account of the areas they come from and the significance of these areas. We have placed the blame on a new set of causes associated principally with the intensity of the sound signal, the size of the speaker's vocal equipment and its health.

We confirmed the work in predicting error prone utterances using a corpus free of side noise or other interference.

6.3.3 Identification of Language Errors

In relation to research into language models, we confirm and extend the work on using out-of-grammar meta-words and phoneme and domain based language models by applying each of these methods in turn to a standard data set. We have carried out a comprehensive comparison of the techniques and shown that meta-words appear to represent a particularly good balance of effort for return in terms of prediction accuracy. In addition, we have identified a function that can be used to model corpora and the incidence of out-of-vocabulary words.

6.4 Future Work

In this section we review a number of the strands of future research that arise from this thesis.

6.4.1 Developing a Full Bayesian Network

Problem: Discriminating between acoustic errors and language errors.

The techniques developed in this thesis are designed to solve a general problem: *Does this hypothesis contain an error?* However, there are obvious advantages in fielded applications being able to establish if the error is in the acoustic domain or in the language domain. A dialogue manager that knows the user is speaking in a fashion that the speech recognizer will never accurately handle can simply pass the user over to an operator. If the problem is the form of the language being employed, suitable advice can be offered to the user and the dialogue can continue automatically.

In theory a Bayesian Network would prove suitable for such a task, as it models the causes of errors in a manner that tries to capture the dependencies between the different causes. In Section 5.3.3 we use a naive Bayes classifier as one of our machine learning techniques. It proved the least successful of our methods with a prediction accuracy of a little under 87% and a Kappa of just over 68%. We pointed out at the time that this poor performance was probably due to the fact that a naive Bayes classifier assumes independence amongst all predictive features. In our case this assumption does not hold; clearly, for example, all the formants for any particular speaker are closely related, as are minimum, maximum and average intensity and a number of other features. We started building a simple Bayesian Network by treating the acoustic and language features as independent and got an immediate improvement in prediction accuracy to over 91% and in Kappa to nearly 78%.

One might reasonably assume that the following groups of features are independent of each other:

- length of utterance,
- the various speech pathology features,
- intensity,
- pitch and formants
- speaking rate features,
- start of speech,
- acoustic score (from the automatic speech recognizer), and
- the language domain metrics.

Based upon this observation one might create and train a Bayesian Network that achieves much higher prediction accuracy, perhaps even at levels in excess of the other methods we used. In addition, such an approach is likely to reveal if the cause of any

error lies in the acoustic or language domain. Knowing this has obvious advantages for a commercial system. If a user is employing unknown language the system can advise them what sort of language they can employ. If the problem is acoustic they can be immediately passed to an operator. This procedure maximizes the proportion of interactions that can be handled automatically, while at the same time minimizing user frustration.

Methods for learning Bayesian networks with both continuous and discrete variables can be found in the literature [Bøttcher, 2001, 2002]. In addition a package for R called DEAL has been written to implement the approach [Bøttcher and Dethlefsen, 2003]. This opens the way to using machine learning to create a Bayesian Network that handles each independent group as a node in a network leading to the outcome. The acoustic and language features would then be likely to both be parents to the outcome node.

6.4.2 Associating Errors with the Vocal Tract

Problem: Pitch is not a feature recognizers use, yet it is still a significant predictor of errors.

We know that variation in pitch and formants are significant in predicting acoustic errors. However, pitch itself cannot directly play a role in acoustic errors as it is largely removed by the various transformations that incoming sound is subjected to before acoustic models are built and before the recognizer endeavors to handle user utterances. Pitch must be closely associated with some feature the recognizer does use. The recognizer is trying to capture the disposition of the vocal tract at the time of the creation of a sound and we know that pitch is also a function of the vocal tract. A hypothesis we developed to account for this, but which we did not test was: *Differences in the length, and cross section, of the vocal tract are associated with errors.*

It is possible to estimate the length of a speaker's vocal tract directly from a sound file using Linear Predictive Coding (LPC). LPC tries to determine the formant frequencies, or peaks in the filter. If one knows the form of the source and the output waveform, one can calculate the properties of the filter that transformed that source into that output. One makes certain assumptions about the general situation of formants in the speech being analyzed. One expects around one formant for every 1000 Hz in adult male speech, every 1100 Hz in adult female speech and under one formant per 2000 Hz in children's speech. LPC analysis seeks to minimize the difference between the predicted (synthesized) signal and the actual signal, revealing the vocal tract length and some information about the vocal tract cross section.

The features of LPC set some limits on the technique. In particular, one must know if one is working with adult or children's male or female speech and the sampling rate.

We experimented with using the technique[1] but found that we could not get reliable estimates of vocal tract length and so abandoned the work. Of course there may be more effective tools than we tried to use to achieve these measurements. Indeed, it might be possible to obtain actual measurements, rather than relying on estimates, by using:

Magnetic resonance imaging, where the vocal tract is tagged by bands of magnetic 'stains' [Napadow et al., 1999];

Ultrasound, where ultrasound equipment is used to capture the size and disposition of the vocal tract [Ostry and Munhall, 1985; Lundberg and Stone, 1999; Stone, 1990; Stone and Lundberg, 1996]; or

X-ray micro-beams, where x-ray beams of a size down to and below one micrometer are used [Adams et al., 1994].

In addition, Dromery et al. [2006] shows that very accurate measurements can be taken of those parts of the vocal equipment that are easily physically accessible using a suitably modified Bio-Research Associates JT-3 jaw tracking instrument.

These methods could allow one to gather accurate estimates of vocal tract length and cross section for speakers and one could then use them in statistical studies, of the sort used in this thesis, to see if there is a significant correlation between vocal tract length and/or cross section with acoustic errors. Techniques already exist to try and automatically compensate for variations in vocal tract length (see Puming and Alex [1997] for three such methods). If the correlation turns out to be significant, such techniques could be focused on with consequent benefits to prediction accuracy levels of speech recognizers.

6.4.3 Improving Corpus Modeling

Problem: Practitioners are unable to estimate the size of a corpus required to provide a certain level of language coverage in fielded systems.

Practitioners use corpora collected in the field to develop and refine systems following the adage that past experience in any domain is the best evidence for future experience. It would be useful to be able to determine the size of corpus required to give a predetermined level of coverage of the language likely to be experienced in a domain in the future. In Section 4.5.3 we noted that the rate new words are introduced into a corpus can be described by the formula:

$$y = \alpha x^{\beta} \tag{6.1}$$

[1] We carried out these experiments using Praat.

where y is the rate of introduction of new words, x is the size of the training set, and α and β are the coefficients required to produce a curve that fits the actual data concerning the rate of introduction of new words in either of the two corpora we studied. If one could establish that this formula held across a wider range of corpora, and the coefficients to be used, one could estimate the size of corpus required to achieve any desired level of language coverage.

To do this one would have to study a number of corpora to see:

- if the formula held over the new data, and
- if there were any relationships between the coefficients required in the different corpora.

In our two corpora β turned out to be similar in both cases at -0.8159 and -0.8688, but α varied between 284.62 and 157.73. Interestingly the greater figure occurred with the corpus having the more complex vocabulary so it might be possible to establish some sort of correlation between vocabulary size and α. If some rational basis for estimating these parameters can be established one could model all the various features of this problem: the training and test set size, and out-of-vocabulary rates using integration:

$$OOV(C_{test}) = \frac{\int_{C_{test}}^{C_{train}} \alpha(C_{test} - C_{train})^{\beta}.d(C_{test} - C_{train})}{C_{test} - C_{train}} \tag{6.2}$$

Equation 6.2 can be rearranged to enable one to find any missing variable given the others are known, allowing one to estimate:

- the size of training set required for a given level of out-of-vocabulary incidents over a future period,
- the level of out-of-vocabulary incidents from a given training set over a future period, and
- the effective future life of a language model given a particular training set and a given ceiling on the rate of out-of-vocabulary incidents.

6.4.4 Application to Extended Speech

Problem: Recognizers can be faced with extended periods of speech when listening to dictation, or meeting room speech.

Mishearing occurs not only when one is faced with short utterances of the sort found in our chosen domain of telephony-based speech applications; they also occur in extended continuous speech, perhaps most typically characterized by dictation systems, but also encompassing speech in meeting rooms and other situations.

Of course, even so-called continuous speech recognition systems have to chunk their sound input into discrete segments for processing. Although these chunks may have a longer average length than the utterances we have dealt with in this thesis they should be susceptible to processing using exactly the same techniques we have used.

It is likely that these sort of systems will use either a recognizer trained for one speaker, for example with dictation, or a recognizer with more general acoustic models, for example for meeting rooms. One of the more interesting aspects of this sort of task is to identify the appropriate forms of actions to be taken by the system when errors occur. A dictation system might model the actions of an old style human shorthand-typist who seeks corrective action as the dictation progresses. The problems are more deep rooted in a meeting room scenario where the system might have to challenge speakers for clarifications as the dialogue progresses, which could affect the very flow of discussions.

6.5 Final Words

This thesis commenced with a review of the problems that arise from errors in spoken language dialogue systems. The relevant parts of that review are reported on in Chapter 2, but the review actually covered the full range of problems that are associated with errors from their first happening when an utterance is misrecognized, through their consequences when they enter dialogues, to the many and various techniques and strategies that have been developed to try to repair them. As the work progressed, it became clear to us that there was a chance to make a real contribution to solving one problem: knowing that the system has misheard. The techniques we used achieved a prediction accuracy of 95.4% in classifying utterances as correctly recognized or not during our assessment.

Bibliography

S. G. Adams, G. Weismer, and R. D. Kent. 1993. *Journal of Speech Hearing Research*, 36:41–54, 1994.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akademiai Kaid.

J. Aldrich. R.A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):162–176, 1997.

J. Allen. Robust understanding in a dialogue system. In *34th Meeting of the Association for Computational Linguistics*, 1995.

J. Austin. *How to do things with words*. Oxford University Press, 1962.

R. H. Baayen. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Dordrecht: Kluwer Academic Publishers, 2001.

L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:179–190, 1983.

B. Balentine and D. Morgan. *How to Build a Speech Recognition Application*. Enterprise Integration Group, San Ramon, California, 1999.

I. Bazzi and J. Glass. A multi-class approach for modelling out-of-vocabulary words. ICSLP, 2002.

I. Bazzi and J. R. Glass. Modeling out-of-vocabulary words for robust speech recognition. pages 401–404. ICSLP-2000, 2000.

C. Bennet and A. I. Rudnicky. The Carnegie Mellon Communicator Corpus. In *Proceedings of ICSLP 2002*, pages 341–344, Denver, Colarado, 2002.

R. Berk. An introduction to ensemble methods for data analysis. Department of Statistics Papers Paper 2005032701, Department of Statistics, UCLA, 2005.

D. Bohus and A. Rudnicky. A K hypotheses + other belief updating model. In *AAAI Workshop on Statistical and Empirical Aproaches for Spoken Dialogue Systems*. AAAI, 2006.

D. Bolinger. *Intonation and Its Uses*. Stanford University Press, 1989.

D. L. Bolinger. A theory of pitch accent in English. *Word*, 14:109 – 149, 1958.

M. Boros, M. Aretoulaki, F. Gallwitz, H. Niemann, and E. Nöth. Semantic processing of out-of-vocabulary words in a spoken dialogue system. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH'97)*, volume 4, pages 1887–1890, Rhodes, Greece, 1997.

S. G. Bøttcher. Learning Bayesian networks with mixed variables. In *proceedings of the Eight International Workshop in Artificial Intelligence and Statistics*, 2001.

S. G. Bøttcher. Learning baysian networks with mixed variables. Technical report, Aalborg University, 2002.

S. G. Bøttcher and C. Dethlefsen. Deal: A package for learning bayesian networks. *Journal of Statistical Software*, 8(20), 2003.

C. Bousquet-Vernhettes, R. Privat, and N. Vigouroux. Error handling in spoken dialogue systems: toward corrective dialogue. In *Error Handling in Spoken Langugage Dialogue Systems*, pages 41 – 45. International Speech Communication Association, 2003.

C. Bousquet-Vernhettes and N. Vigouroux. Recognition error handling by the speech understanding system to improve spoken dialogue systems. In *Error Handling in Spoken Langugage Dialogue Systems*. International Speech Communication Association, 2003.

L. Breiman. Predicting games and arcing algorithms. Technical report, Technical report 504, Statistics Department, University of California, 1997. Submitted to Neural Computing.

L. Breiman. Random Forests. *Machine Learning*, 45(1):5–33, 2001.

L. Breiman. *Manual On Setting Up, Using, And Understanding Random Forests V3.1*. University of California, Berkey Campus, 2002.

L. Breiman, J. H. Friedman, and R. A. Olshen. *Classification and Regression Trees*. Wadsworth, 1994.

L. Brieman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.

P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roosin. A statistical approach to machine transaltion. *Computational Linguisitics*, 16(2):79–85, June 1990.

C. G. Broyden. The convergence of a class of double rank minimization algorithm parts i and ii. *J. of Institute of Mathematics and its Applications*, 6, 1970.

I. Bulyko, M. Kirchhoff, M. Ostendorf, and J. Goldberg. Error correction detection and response generation in a spoken dialogue system. *Speech Communication*, 45 (271–288), 2005.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121 – 167, 1998.

K. P. Burnham and D. R. Anderson. Kullback-Leibler information as a basis for strong inference in ecological studies. *Wildlife Research*, 28:111–119, 2001.

K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: a practical information-theoretic approach*. Springer-Verlag, New York, 2 edition, 2002.

J. Carletta. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

P. Carpenter, C. Jin, D. Wilson, R. Zhang, D. Bohus, and A. Rudnicky. Is this conversation on track? In *Proceedings of Eurospeech 2001*, pages 2121–2124, Aalborg, Denmark, 2001. Eurospeech.

S. Cassidy. Comp449: Speech Recognition. http://www.ics.mq.edu.au/ cassidy/comp449/html/index.html, 2002.

L. L. Chase. *Error-Responsive Feedback Mechanisms for Speech Recognizers*. PhD thesis, Robotics Institute, Carnegie-Mellon University, 1997.

K. Chen and M. Hasegawa-Johnson. How prosody improves word recognition. Nara, Japan, March 2004. Speech Prosody, ISCA.

H. Chih-Wei, C. Chih-Chung, and L. Chih-Jen. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2003.

A. Chotimongkol. Improving speech recognizer performance in a dialogue system using n-best hypotheses reranking. Master's thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2001.

S. Choularton and R. Dale. User responses to speech recognition errors: Consistency of behaviour across domains. Sydney, December 2004. Australian Language Technology Workshop, ASSTA.

P. R. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge Toolkit. ESCA Eurospeech, 1997.

J. Cohen. Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. *Journal of the Acoustic Society of Amnerica*, 97(5):3246–3247, May 1995.

W. Cohen. Learning trees and rules with set-valued featurers. In *AAAI-96*, 1996.

T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley and Sons, New York, New York, 1991.

H. Cuayáhuitl and B. Serridge. Out-of-vocabulary word modeling and rejection for Spanish keyword spotting systems. In *Lecture Notes In Computer Science*, pages 156–165, London, 2002. Second Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, Springer-Verlag.

W. Daelemans and V. Hoste. Evaluation of machine learning methods for natural language processing tasks. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 755–760, Los Palmas, Gran Canaria, 2002.

M. Danieli. On the use of expectations for detecting and repairing human-machine miscommunication. *Computational Linguistics*, 13:11–24, 1996.

S. Das, D. Nix, and M. Pichney. Improvements in children's speech recognition performance. Seattle, W.A., May 1998. ICASSP.

B. Decadt, J. Duchateau, W. Daelemans, and P. Wambacq. Memory-based phoneme-to-grapheme conversion. In A. N. M. Theune and H. Hondrop, editors, *Proceedings of the Twelfth Meeting of Computational Linguistics in the Netherlands (CLIN 2001)*, number 45 in Language and Computers: Studies in Practical Linguistics, pages 47–61, Amsterdam, December 2002a. Rodopi. URL `http://www.cnts.ua.ac.be/Publications/2002/DDDW02`.

B. Decadt, J. Duchateau, W. Daelemans, and P. Wambacq. Transcription of out-of-vocabulary words in large vocabulary speech recognition based on phoneme-to-grapheme conversion. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, volume 1, pages 861–864, Orlando, Florida, U.S.A, May 2002b. URL `http://www.cnts.ua.ac.be/Publications/2002/DDDW02a`.

E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2004. R package version 1.5-5.

G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. Sheeps, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proc. of Intl. Conf. on Speech and Language Processing*. ICSLP, 1998.

G. R. Doddington and T. B. Schalk. Speech recognition: Turning theory to practice. *IEEE Spectrum*, 18(9), 1981.

P. Domingoes and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–137, 1997.

C. Dromery, S. Nissen, P. Nohr, and S. G. Fletcher. Measuring tongue movements during speech: Adaptation of a magnetic jaw-tracking system. *Speech Communication*, 48: 463–474, 2006.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, N.Y., 2001.

K. A. Ericsson and P. F. Delaney. *Working Memory and Thinking*, chapter Working Memory and Expert Performance., pages 93–114. Psychology Press, 1999.

A. Facco, D. Falavigna, R. Gretter, and M. Vigana. Design and evaluation of acoustic and language models for large scale telephone services. *Speech Communication*, 48 (2):176–190, 2006.

G. Fant. *Acoustic Theory of Speech Production*. The Hagues, Mounton, 1960.

G. Fant. Glottal flow: models and interaction. *Journal of Phonetics*, (14):393–399, 1986.

R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13: 317–322, 1970.

Y. Freund and R. E. Schapire. Experiments with a new boosting algortihm. In *Machine learning: Proceedings of the Thirteenth International Conference*, pages 148–156, San Fransciso, 1996. Morgan Kaufman.

Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999. Appearing in Japanese, translation by Naoki Abe.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–374, 2000.

D. B. Fry. Duration and intensity as physical correlates of linguistic stress. *Journal of Acoustical Society of America*, 30:765 – 769, 1955.

D. B. Fry. Experiments in the perception of stress. *Language and Speech*, 1:126 – 152, 1958.

M. Gabsdil. Classifying recongition results for spoken dialog systems. Sapporo, Japan, July 2003. ACL Students Workshop.

F. Gallwitz, E. Noeth, and H. Niemann. A category based approach for recognition of out-of-vocabulary words. In *Proc. ICSLP '96*, volume 1, pages 228–231, Philadelphia, PA, 1996. URL `citeseer.ist.psu.edu/gallwitz96category.html`.

R. Gentleman, V. Carey, W. Huber, R. A. Irizarry, and S. Dudoit. *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Springer, 2005.

L. Gillick, Y. Ito, L. Manganaro, M. Newman, F. Scattone, S. Wegmann, J. Yamron, and P. Zhan. Dragon Systems' Automatic Transcription of the New TDT Corpus. Technical report, DARPA, 1998.

D. Giuliani and M. Gerosa. Investigating recognition of children's speech. ICASSP, April 2003.

D. Goldfarb. A family of variable-metric methods derived by variational means. *Maths. Comp.*, 24:23–26, 1970.

G. Gorrell. Using statistical language modelling to identify new vocabulary in a grammar-based speech recognition system. Eurospeech, 2003.

G. Gorrell. Language modelling and error handling in spoken dialogue systems. Licentiate, Graduate School of Language Technology, Faculty of Arts, Göteborg University, Göteborg, Sweden, 2004.

G. Gorrell, I. Lewin, and M. Rayner. Adding intelligent help to mixed-initiative spoken dialogue systems. ICSLP-2002, 2002.

P. Green, J. Carmichael, A. Hatzis, P. Enderby, M. Hawley, and M. Parker. Automatic speech recognition with sparse training data for dysarthric speakers. Technical report, University of Sheffield, UK, 2003.

S. Greenberg, S. Chang, and J. Hollenback. An introduction to the Diagnostic Evaluation of Switchboard-Corpus Automatic Speech Recognition Systems. In *NIST Speech Transcription Workshop, College Park, MD*. NIST, May 2000.

H. P. Grice. *Syntax and Semantics*, volume 3, pages 43–48. Academic Press, New York, 1975.

R. Grishman. Information extraction and speech recognition. Technical report, Computer Science Department, New York University, 1998.

B. J. Grosz and C. L. Sidner. *Intentions in Communication*, chapter Plans for Discourse, pages 417 – 444. MIT Press, Cambridge, MA, 1990.

I. Gurevych and R. Porzel. Using knowledge-based scores for identifying best speech recognition hypothesis. In *Error Handling in Spoken Langugage Dialogue Systems*. International Speech Communication Association, 2003.

L. Q. Ha, Sicilia-Garcia, J. Ming, and F. J. Smith. Extension of zipfs law to words and phrases. In *The 17th International Conference on Computational Linguistics*, 2002.

K. Hacioglu, B. Pellom, and W. Ward. Parsing speech into articulatory events. icassp, 2004.

K. Hadding-Koch. Acoustico-phonetic studies in the intonation of southern Swedish. Technical report, C. W. K. Gleerup, Lund, Sweden, 1961.

A. Hagen, B. Pellom, and R. Cole. Children's speech recognition with application to interactive books and tutors. St. Thomas, USA, 2003. IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop.

J. D. Harnsberger and L. A. Goshert. Reduced, citation, and hyperarticulated speech in the laboratory: Some acoustic analyses. Progress Report 24, Speech Research Laboratories, Department of Psychology, Indiana University, Bloomington, IN, 2000.

J. Harrington and S. Cassidy. *Techniques in Speech Acoustics*. Klewer, 2000.

M. Hasegawa-Johnson, K. Chen, J. Cole, S.-S. K. Sarah Borys, A. Cohen, T. Zhang, J.-Y. Choi, H. Kim, and S. C. Taejin Yoon. Simultaneous recognition of words and prosody in the Boston University Radio Speech Corpus. *Speech Communication*, 46: 418–439, 2005.

T. Hastie, R. Tibshirani, and J. H. Freeman. *The Elements of Statistical Leaning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, 2001.

T. Hazen and I. Bazzi. A comparison and combination of methods for oov word detection and word confidence scoring. ICASSP, 2001.

R. Higashinaka, K. Sudoh, and M. Nakano. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. *Speech Communication*, 48:417–436, 2006.

J. Hirschberg, D. Litman, and M. Swerts. Prosodic cues to recognition errors. In *ASRU*, 1999.

J. Hirschberg, D. Litman, and M. Swerts. Generalizing prosodic prediction of speech recognition errors. In *International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, September 2000. ICSLP.

J. Hirschberg, D. Litman, and M. Swerts. Prosodic and other cues to speech recognition failures. *Speech Communication*, (43):155–175, 2004.

B. A. Hockey, J. Dowding, G. Aist, and J. Hieronymous. Targeted help and dialogue about plans. Philadelphia, 2002. ACL-02.

E. Horvitz and T. Paek. Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems. In *Eighth Conference on User Modeling*, Sonthofen, Germany, July 2001.

G. Hripcsak and A. S. Rothschild. Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Informatic Association*, 12(3):296 – 8, May – Jun 2005.

X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Prentice Hall PTR, Upper Saddle River, New Jersey, 2001.

I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, 2002.

C. Joon-Ki and O. Yung-Hwan. N-gram adaptation with dynamic interpolation coefficient using information retrieval technique. *IEICE - Transactions on Information and Systems*, E89-D(9):2579–2582, September 2006.

T. Kawabata and M. Tamoto. Back-off method for n-gram smoothing based on binomial posterioridistribution. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing.*, volume 1, pages 192–195, Atlanta, May 1996. IEEE.

M. Kehoe, C. Stoel-Gammon, and E. H. Buder. Acoustic correlates of stress in young children's speech. *Journal of Speech and Hearing Research*, 38:338 – 350, 1995.

J. Koreman. *Decoding Linguistic Information in the Glottal Airflow.* PhD thesis, University of Nijmege, 1996.

A. Kornai. How many words are there? *Glottometrics*, 4:61–68, 2006.

E. Krahmer, M. Swerts, M. Theune, and M. Weegels. Error Detection in Spoken Human-Machine Interaction. *International Journal of Speech Technology*, 4(1):19 – 23, 2001.

K. Krippendorf. *Content Analysis: An introduction to its methodology.* Sage Publishing, 1980.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

P. Kwok. Rejecting out-of-grammar utterances. Internet, June 2004. URL `http://www.speech.cs.cmu.edu/sphinx/twiki/bin/view/Sphinx4/RejectionHandling`.

S. Lemmety. Review of speech synthesis technology. Master's thesis, Helsinki University of Technology, 1999.

V. I. Levenshtein. Binary codes capable of correcting insertions and reversals. *Sov. Phys. Dokl*, 10:707–710, 1966.

G. Levow. Characterizing and Recognizing Spoken Corrections in Human-Computer Dialogue. In *COLING-ACL '98*, Montreal, Canada, 1998. COLING-ACL.

G. Levow. Understanding recognition failures in spoken corrections in human-computer dialogue. In *ESCA Workshop on Dialogue and Prosody*, Eindhoven, Netherlands, 1999. ESCA.

Q. Li and M. J. Russell. An analysis of the causes of increased error rates in children's speech recognition. Denver, CO., 2002. ICSLP.

L. Libuda. Improving clarification dialogs in speech command systems with the help of user modelling: A conceptualization for an in-car user interface. ABIS-Workshop 2001, 2001.

P. Lieberman. Some acoustic correlates of word stress in American-English. *Journal of Acoustical Society of America*, 32:451 – 454, 1960.

D. J. Litman, J. Hirschberg, and M. Swerts. Predicting User Reactions to System Error. In *Meeting of the Association for Computational Linguistics*, pages 362–369, Toulouse, France, 2001.

R. López-Cózar and Z. Callejas. Combining language models in the input interface of a spoken dialogue system. *Computer Speech and Language*, 20(4):420–440, 2006.

A. J. Lundberg and M. Stone. Three-dimensional tongue surface reconstruction: practical considerations for ultrasound data. *Journal of the Acoustic Society of America*, 106:2858–2867, 1999.

K. Maekawa. Phonetic and phonological characteristics of paralinguistic information in spoken Japanese. In *Proceedings of the International Conference on Spoken Language Processing*, 1998.

A. Martin and M. Przybocki. The NIST 1999 speaker recognition evaluation - an overview. Technical report, National Institute of Standards and Technology, 1999.

M. J. Mazerolle. *Mouvements et reproduction des amphibiens en tourbires perturbes*. PhD thesis, Faculté de Foresterie et de Gomatique, Université Laval, 2004.

D. McNeill. *Epidemiological Research Methods*. John Wiley & Sons, 1998.

S. W. McRoy and G. Hirst. The repair of speech act misunderstanding by abductive inference. *Computation Linguistics*, 21(4):435–478, 1995.

M. McTear, I. A. O'Neill, P. Hanna, and X. Liu. Handling errors and determining confirmation strategiesan object-based approach. *Speech Communication*, 45(249–269):83–86, 2005.

R. Meir and G. Rätsch. *An introduction to boosting and leveraging*, chapter Advanced Lectures on Machine Learning, pages 119–184. Springer, 2003.

R. Meliani and D. O'Shaughnessy. New efficient fillers for unlimited word recognition and keyword spotting. In *ICSLP'96*, Philadelphia, Pennsylvania, USA, 1996. ICSLP. URL `citeseer.ist.psu.edu/meliani96new.html`.

E. Mengusoglu and C. Ris. Use of acoustic prior information for confidence measure in ASR applications. Scandinavia, 2001. Eurospeech 2001.

MIT. Jupiter (1–888–573–TALK). http://groups.csail.mit.edu/sls//applications/jupiter.shtml, 2006.

S. Molau, S. Kanthak, and H. Ney. Efficient vocal tract normalization in automatic speech recognition. In *In Proc. of the ESSV00*, pages 209–216, 2000.

J. J. M. Monaghan, C. Feldbauer, T. C. Walters, and R. D. Patterson. Low-dimensional, auditory feature vectors that improve vocal-tract-length normalization in automatic speech recognition. pages 477–482, Paris, 2008. Acoustics 08.

J. Müller, H. Stahl, and M. Lang. Predicting the out-of-vocabulary rate and the required vocabulary size for speech processing applications. Technical report, Institute for Human-Machine-Communication, Munich University of Technology, Munich, 1995. URL `citeseer.ist.psu.edu/247335.html`.

H. Nanjo, A. Lee, and T. Kawahara. Automatic diagnosis of recognition errors in large vocabulary continuous speech recognition systems. In *ICSLP*, volume 2, pages 1027 – 1030, 2000.

V. J. Napadow, Q. Chen, V. J. Wedeen, and R. J. Gilbert. Intramural mechanics of the human tongue in association with physiological deformations. *Journal of Biomechanics*, 32:1–12, 1999.

H. Nyquist. Certain topics in telegraph transmission theory. *AIEE*, 47:617–644, April 1928.

S. Öhgren. Experiment with adaptive and vocal trat length normalization at automatic speech recognition of children's speech. Masters, Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, 2007.

M. Orlandi, C. Culy, and H. Franco. Using dialog corrections to improve speech recognition. In *Error Handling in Spoken Langugage Dialogue Systems*. International Speech Communication Association, 2003.

D. J. Ostry and K. G. Munhall. Control of rate and duration of speech movements. *Journal of the Acoustic Society of America*, 77:640–648, 1985.

S. Oviatt, M. MacEachern, and G. Levow. Prediciting Hyperarticulate Speech During Human-Computer Error Resolution. *Speech Communication*, 24(2):87–110, 1998.

S. L. Oviatt. Predicting spoken disfluencies during human-computer interaction. *Computer Speech and Language*, 9:19–35, 1995.

D. D. Palmer and M. Ostendorf. Improving information extraction by modeling errors in asr output. In *the Human Language Technology Workshop*, pages 156–160, March 2001.

R. Patel and D. Roy. Teachable interfaces for individuals with dysarthric speech and severe physical disabilities. Technical report, Department of Speech-Language Pathology, University of Toronto and The Media Laboratory, Massachusetts Institute of Technology, 1998.

B. Pellom. SONIC: The University of Colorado Continuous Speech Recognizer. Technical Report TR-CSLR-2001-01, University of Colorado, 2001.

F. Peng. The sparse data problem in statistical language modeling and unsupervised word segmentation. PhD proposal, citeseer.ist.psu.edu/489036.html, 2001.

D. Perlis and K. Purang. Conversational Adequacy: Mistakes are the Essence. Technical report, Department of Computer Science, University of Maryland, 1996.

J. Pinto, H. Bourlard, Z. D. Greve, and H. Hermansky. Comparing different word lattice rescoring approaches towards keyword spotting. Technical report, IDIAP Research Institute, Martigny, Switzerland, July 2007.

J. Pitrelli, C. Fong, S. Wong, J. Spitz, and H. Leung. Phonebook: a phonetically rich isolated word telephone speech database. Detroit, 1995. ICASSP'95.

M. Pitz, S. Molau, R. Schlter, R. S. Uter, and H. Ney. Vocal tract normalization equals linear transformation in cepstral space. In *in Proc. of the EUROSPEECH01*, pages 2653–2656, 2001.

K. E. Pollock, D. M. Brammer, and C. F. Hageman. An acoustic analysis of young childrens productions of word stress. *Journal of Phonetics*, 21:183 – 203, 1990.

A. Pontamianos and S. Narayanan. Robust recognition of children's speech. In *Transactions on Speech and Audio Processing*. IEEE, 2003.

A. Pontamianos, S. Narayanan, and S. Lee. Automatic speech recognition for children. Rhodes, Greece, September 1997. EUROSPEECH.

P. Prodanov and A. Drygajlo. Bayesian networks based multimodality fusion for error handling in human robot dialogues under noisy conditions. *Speech Communication*, 45:211–29, 2005.

Z. Puming and W. Alex. Vocal tract length normalization for large vocabulary continuous speech recognition. Technical Report CMU-CS-97-148, Carnegie-Mellon University, Pittsburg, May 1997. URL `citeseer.ist.psu.edu/zhan97vocal.html`.

K. Purang. *Systems that detect and repair their own mistakes*. PhD thesis, University of Maryland, 2001.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Princeton HalL, 1993.

M. W. C. Reynaert. *Text-Induced Spelling Correction*. PhD thesis, University of Vrijdag, 1963.

E. Ringger. *Correcting Speech Recognition Errors*. PhD thesis, University of Rochester, 2000.

B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

I. Rish. An empirical study of the naive bayes classifier. In *Workshop on Empirical Methods in Artificial Intelligence*. IJCAI, 2001.

B. Roark. Markov parsing: lattice rescoring with a statistical parser. In *Proceedings of the 40th Annual Meeting of the Associtation for Computational Lingusitics (ACL).*, pages 287–294, Philladelphia, July 2002. ACL.

M. Rotaru and D. J. Litman. Discourse structure and speech recognition problems. Pittsburg, 2006. Interspeech06.

M. Rothenberg. Some relations between glottal air flow and vocal fold contact area. *ASHA Reports*, (11):88–96, 1981.

R. San-Segundo, B. Pellom, and W. Ward. Confidence measures for dialogue management in the CU Communicator System. In *ICASSP'2000*, 2000.

S. Saraswathi and T. Geetha. Time scale modification and vocal tract length normalization for improving the performance of tamil speech recognition system implemented using langage independent segmentation algorithm. *International Journal of Speech Technology*, 2008.

N. Sawhney and S. Wheeler. Using phonological context for improved recognition of dysarthric speech. Technical report, Speech Interface Group, MIT Media Lab, 1999.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

R. E. Schapire. A brief introduction to boosting. Sixteenth International Joint Conference on Artificial Intelligence, 1999.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2001.

J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, 1969.

J. R. Searle. *Expression and Meaning*. Cambridge University Press, Cambridge, 1979.

D. F. Shanno. Conditioning of quasi-Newton method for function minimization. *Maths. Comp.*, 24:647–656, 1970.

C. Shin and G. Kochanski. Prosody and prosodic models. Denver, Colarado, September 2002. ICSLP.

J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd. Analysis of User Behavior under Error Conditions in Spoken Dialog. In *Proc. ICSLP*, pages 2069 – 2072, Denver, Colorado, 2002.

E. Shriberg and A. Stolcke. Prosody Modeling for Automatic Speech Understanding: An Overview of Recent Research at SRI. In *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, pages 13 – 16. ISCA, Red Bank, NJ., 2001.

V. Siivola, T. Hirsiäki, M. Creuts, and M. Kurimo. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. pages 2293 – 2296, Geneva, 2003. EUROSPEECH 2003.

K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. ToBI: A standard for labeling english prosody. In *ICSLP*, pages 867–870, 1992.

G. Skantze. Exploring human error handling strategies: Implications for spoken dialogue systems. In *Error Handling in Spoken Langugage Dialogue Systems*. International Speech Communication Association, 2003.

G. Skantze. Exploring human error recovery strategies: Implications for spoken dialogue systems. *Speech Communication*, 45:325–341, 2005.

A. M. C. Sluijter and V. J. van Heuven. Spectral balance as an acoustic correlate of linguistic stress. *Journal of Acoustical Society of America*, 100(4):2471–2485, 1996.

A. M. C. Sluijter, V. J. van Heuven, and J. J. A. Pacilly. Spectral balance as a cue in the perception of linguistic stress. *Journal of Acoustical Society of America*, 101(1): 503–513, 1997.

D. R. R. Smith, R. D. Patterson, R. Turner, H. Kawahara, and T. Irino. The processing and perception of size information in speech sounds. *Journal of the Acoustic Society of America*, 117:305–318, 2005.

H. Soltan and A. Waibel. On the influence of hyperarticulated speech on the recognition performance. In *International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

K. N. Stevens. *The Handbook of Phonetic Sciences*, chapter Articulatory-Acoustic-Auditory Relationship. Blackwell Publishers Ltd, Oxford, 1997.

L. J. Stifelman. User Repairs of Speech Recognition Errors: An Intonational Analysis. Technical report, Speech Research Group, MIT Media Laboratory, May 1993.

M. Stone. A three-dimensional model of tongue movement based on ultrasound and x-ray microbeam data. *Journal of the Acoustic Society of America*, 87:2207–2217, 1990.

M. Stone and A. Lundberg. Three-dimensional tongue surface shapes of English consonants and vowels. *Journal of the Acoustic Society of America*, (3728–3737), 1996.

F. Tezuka, T. Namiki, and H. Higashiiwai. Observer variability in endometrial cytology using kappa statistics. *Journal of Clinical Pathology*, 45(4):292–294, April 1992.

I. R. Titze. *Principles of Voice Production*. Prentice Hall, Englewood Cliffs, 1996.

D. R. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, University of Rochester, New York, 1994.

R. G. Tull and J. C. Rutledge. "Cold Speeech" for automatic speaker recognition. Honolulu, Hawaii, December 1996. 3rd Joint ASA/ASJ Meeting.

A. E. Turk and J. R. Sawusch. The processing of duration and intensity cues to prominence. *Journal of Acoustical Society of America*, 99(6):3782–3790, 1996.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, 4 edition, 2002.

M. Walker, J. Aberdeen, J. Boland, E. Braat, J. Garofolo, L. Hirschman, A. Lee, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stollard, and S. Whittaker. DARPA Communicator Dialogue Travel Planning Systems: The June 2000 Data Collection. Aalborg, Denmark, 2001. Eurospeech.

M. Wang and J. Hirschberg. Automatic classification of intonational phrase boundaries. *Computer Speech and Language*, 1992.

Z. Wang, P. Ding, and B. Xu. Some issues on the study of vocal tract normalization. Technical report, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 2008.

L. Wellington, H. Ney, and S. Kanthak. Speaker adaptive modeling by vocal tract normalization. *IEEE transactions on speech and audio processing*, 10(6):414–426, 2002.

F. Wessel, K. Macherey, and R. S. Uter. Using word probabilities as confidence measures. In *in Proc. ICASSP*, pages 225–228, 1998.

D. A. G. Williams. *Knowing What You Don't Know: Roles for Confidence Measures in Automatic Speech Recognition*. PhD thesis, Department of Computer Science, University of Sheffield, 1999.

J. Wilpon and C. Jacobsen. A study of speech recognition for children and the elderly. In *In Proceedings of the IEEE Internation Conference on Acoustics, Speech, and Signal Processing*, Atlanta, 1996. IEEE.

D. Wolpert and W. G. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

S. Yildirim and S. S. Narayanan. An information-theoretic analysis of developmental changes in speech. Hong Kong, April 2003. ICASSP.

R. Zhang and A. I. Rudnicky. Word level confidence annotation using combinations of features. In *Proceedings of Eurospeech 2001*, pages 2105–2108, Aalborg, Denmark, 2001.

L. Zhou, J. Feng, A. Sears, and Y. Shi. Applying naive bayes classifier to assist users in detecting speech recognition. IEEE, 2005.

G. K. Zipf. *The psycho-biology of language: an introduction to dynamic philology.* The M.I.T. Press, Cambridge, MA, 2 edition, 1935.

I. Zitoumi and Z. Qiru. Hierarchical linear discounting class n-gram language models: A multilevel class hierarchy approach. In *Conference on Acoustics, Speech and Signal Processing, 2008.*, pages 4917–4920, Los Angelis, March 2008. IEEE.

T. Zollo. Using grammatical analysis to detect misrecognitions. In *Error Handling in Spoken Langugage Dialogue Systems.* International Speech Communication Association, 2003.

E. Zoltan-Ford. How to get people to say and type what computers can understand. *International Journal of Man-Machine Studies*, 34(4):527–547, 1991.