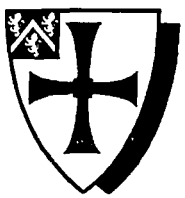
Identification and correction of speech repairs in the context of an automatic speech recognition system

Johnson, Kevin

University of Durham



Identification and Correction of Speech Repairs in the Context of an Automatic Speech Recognition System.

Kevin Johnson

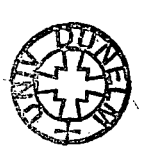
Laboratory for Natural Language Engineering
Department of Computer Science

First Submitted October 1996

Passed June 1997

Submitted in partial fulfilment of the
requirements for the degree of

Doctor of Philosophy



This thesis is dedicated to my Mother and Father.

Abstract

Recent advances in automatic speech recognition systems for read (dictated) speech have led researchers to confront the problem of recognising more spontaneous speech. A number of problems, such as disfluencies, appear when read speech is replaced with spontaneous speech. In this work we deal specifically with what we class as speech-repairs.

Most disfluency processes deal with speech-repairs at the sentence level. This is too late in the process of speech understanding. Speech recognition systems have problems recognising speech containing speech-repairs. The approach taken in this work is to deal with speech-repairs during the recognition process.

Through an analysis of spontaneous speech the grammatical structure of speech-repairs was identified as a possible source of information. It is this grammatical structure, along with some pattern matching to eliminate false positives, that is used in the approach taken in this work. These repair structures are identified within a word lattice and when found result in a SKIP being added to the lattice to allow the *reparandum* of the repair to be ignored during the hypothesis generation process. Word fragment information is included using a sub-word pattern matching process and *cue phrases* are also identified within the lattice and used in the repair detection process.

These simple, yet effective, techniques have proved very successful in identifying and correcting speech-repairs in a number of evaluations performed on a speech recognition system incorporating the repair procedure. On an un-seen spontaneous lecture taken from the Durham corpus, using a dictionary of 2,275 words and phoneme corruption of 15%, the system achieved a correction recall rate of 72% and a correction precision rate of 75%.

The achievements of the project include the automatic detection and correction of speech-repairs, including word fragments and *cue phrases*, in the sub-section of an automatic speech recognition system processing spontaneous speech.

Acknowledgements

I would like thank Professor Roberto Garigliano and Dr. Russell J. Collingham without whom this project would not have been possible. Thanks must also go to EPSRC and the Speech Research Unit of the DRA Malvern, especially Roger Moore and Sue Browning, for helping me through these years.

I have seen many friends and colleagues come and go through the years it has taken me to complete this project. All have contributed to this work in one way or another. There are too many to mention so here are just a few special cases.

Matt Jolly who has laughed with me when I needed it and laughed at me when I didn't. Dave Nettleton who took a distinct dislike to my English when proof reading this thesis and who sat with me on many occasions talking about football (honest!). Jim Bradford who kept reminding me that life could be much worse. Simon Shiu who, despite not being a mate, spent many a time in the coffee room with me thinking of how things could have been and even spent Christmas Eve away from his mates. Agnieszka and Sanjay simply for their support.

Pamela Hatton who has shown me that there is life during & after a PhD project and that it is all worth the effort.

Dr Rick Morgan deserves a special mention for his support in the closing stages of this project

My family must also get a mention for not standing in my way and putting up with my, sometimes odd, sense of personal direction.

A special thank you must go to Uncle Les who always believed in my abilities and gave me encouragement in my endeavours. One who will be sadly missed.

Declaration

The material contained within this thesis has not previously been submitted for a degree at the University of Durham or any other university. The research reported within this thesis has been conducted by the author unless indicated otherwise.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides an introduction to the work presented in this thesis. It shows where in the field of computer science the work fits, the research method used and how the success of the research is to be measured.

1.1 Methodological Issues

The work presented in this thesis has been carried out within the branch of computer science known as artificial intelligence.

Artificial intelligence has, as many would say, become a science in its own right but its origins lie in computer science. Artificial intelligence deals with attempting to simulate human behaviour or more importantly, making a machine do what is normally seen as a human process. At present one of the main aims of artificial intelligence is to mimic the human language process.

The particular area of artificial intelligence in which this work is based is natural language processing. The main approach to natural language processing was to identify or hypothesise as to how humans performed the task of language production & recognition and to mimic this process using machines. The current approach taken by natural language engineers, under the banner of "Natural Language Engineering", is to concentrate more on a "black box" scheme. Here the

input is known, the output is known and a process is formulated that will produce the required output from the known input. This pragmatic approach has become accepted within the artificial intelligence community as the most likely to produce large scale systems, with acceptable performances. Though the most promising approach would be to join the two sides of computational linguistics and natural language engineering, to share knowledge and interchange ideas.

Natural language itself has many possible parts. The three main sections are:

natural language understanding - where the aim is, given a piece of natural language, to decipher the exact meaning of what is being said. This would allow natural dialogues to be performed with machines and would open many possible applications of the new technology.

speech recognition - where the aim is to recognise speech, from a human speaker. This would open technologies developed to anyone who can speak.

speech synthesis - where the aim is to get a machine to speak at a level that can be understood easily by human listeners.

One aim of artificial intelligence and natural language processing research would be to join speech recognisers, synthesisers and natural language understanding processes to give the ultimate "Natural Language System" that can listen, understand and respond either by performing an action or returning speech. Another aim is to produce a system with which people could hold a normal conversation without knowing that they are talking to a machine (as shown partly by the "Turing Test"[1]).

1.1.1 Speech Recognition

In November 1971 DARPA (Defence Advanced Research Projects Agency), currently known as ARPA (Advanced Research Projects Agency), initiated a five year

[1]The "Turing Test" deals with typed text but the ultimate goal would be to speak to a machine.

research and development program which heralded the start of, what is now known as, speech recognition research [Klatt, 1977]. The objectives and constraints of the program meant that the findings were of little use in practical terms. It was not until the advancement of technology in the early eighties that speech recognition became a reality. Systems first came into use as early as 1983 but they were costly and limited in their use. Speech recognition did not become commercially viable until the mid to late eighties. From this start, speech recognition has become a major area of artificial intelligence.

Speech recognition is a very large and complex area of research. Many systems have been produced for specific tasks, with substantial limitations being imposed on the system. However, by decreasing the complexity of the system the solution requirement also decreases allowing a viable solution to be produced. This solution, though adequate for the task in hand, does not normally transfer onto more complex problems. Limitations such as speaker dependence, limited vocabulary and restricted grammar resulted in systems which can rely solely on phonetic and syntactic knowledge. The use of much higher level knowledge such as dialogue, semantics and prosody was not necessary.

As speech recognition was used for more complex problems it was realised that automatic speech recognition systems had reached their performance limit. Therefore, expansions to current systems have been necessary to deal with the problems faced in these more complex tasks. It is here, at the drive for expanding current automatic speech recognition systems to be used for more complex applications where this work lies.

These expansions have moved speech recognition systems from recognising isolated words, to recognising continuous read speech and finally to recognising natural spontaneous speech. Certain problems arise with the introduction of spontaneous speech which are not present with read speech, of which speech disfluencies are one such example. The work described in this thesis investigates speech disfluencies and in particular self-repairs, a form of speech disfluency, which are an integral part of spontaneous speech.

This work builds upon the work presented in [Collingham, 1994] and attempts to produce a practical solution to self-repairs (from now on known as repairs) using the current technology and knowledge that is available within the system.

1.2 Method

The approach taken was to carry out an empirical investigation into repairs and to produce a solution based on the findings. The progression of the thesis almost mimics the schedule of the processes carried out in the investigation. One of the first tasks was to investigate the strengths and weaknesses of the sub-system developed by Collingham (1994) for general speech and speech containing repairs. Data was collected on the style of speech required, lectures/spontaneous monologues, as there was little relevant data in the corpora available at the start of the project. The data collected was then transcribed and analysed with respect to the repairs present in the data. This analysis was compared to an additional analysis performed on another form of speech (human-machine dialogue). This allowed the findings of the analysis of the first set of data to be backed up by findings from a different source of speech (dialogue) and a large number of different speakers (over 100) with different speaking styles. The findings were then formulated into a solution which was coded into the current system. The results were then collected and the success of the solution measured.

1.3 Criteria for Success

Identifying the criteria for success of a project of this type is not an easy task as success can be defined in many ways. Normal criteria for success within the production of a speech recognition system includes such things as robustness, integration, feasibility and usability. In this project we are looking at a single part of the speech recognition process which requires accurate measurements to identify its performance and therefore its success.

Identifying the accuracy of a speech recognition system is normally done through measurements such as "words correct", "word error" and "word accuracy". Where:

words correct - is the percentage of words in the original transcript actually identified by the system.

word error - is the percentage of the number of incorrect words found in the output compared to the number of words in the original transcript.

word accuracy - is the percentage of the number of correct words in the output compared to the number of words in the original transcript.

So that:

$$\begin{aligned} number\ of\ words\ in\ correct\ transcription &= w \\ number\ of\ word\ substitution\ errors &= s \\ number\ of\ word\ deletion\ errors &= d \\ number\ of\ word\ insertion\ errors &= i \\ \%\ words\ correct &= 100 \cdot \frac{w-(s+d)}{w} \quad (1.1) \\ \%\ word\ error &= 100 \cdot \frac{s+d+i}{w} \quad (1.2) \\ \%\ word\ accuracy &= 100 \cdot \frac{w-(s+d+i)}{w} \quad (1.3) \end{aligned}$$

These measurements are acceptable for processing read speech where the exact spoken words need to be recognised, but are not acceptable for spontaneous speech which includes speech disfluencies. These disfluencies need to be removed to allow the true meaning of the utterance to be found. If the speaker said:

The house the Browns house is is the one with the red blue door blue windows.

Here only ten of the original sixteen words spoken are actually required. If the system recognised the true meaning of what the speaker was saying, to be:

The Browns house is the one with the blue windows.

then we can see that the system would be 100% correct as far as the meaning is concerned, but the word error rate would be 37.5%, the words correct rate would be 62.5% and the word accuracy would be 62.5%. Therefore, conventional accuracy measurements for speech recognition systems dealing with spontaneous speech and in-particular dealing with repaired speech are unsuitable.

One measuring scheme used within the speech disfluency field is the "recall" and "precision" rates of the disfluency processes. Here the number of actual disfluencies are compared to the number of disfluencies identified and corrected in the output. This gives the following four measurements:

Recognition Recall - gives the number of correctly identified repairs as a percentage of the number of repairs present in the text. Therefore, if the system found 25 of the 50 repairs in the original passage then the recognition recall rate would be 50%.

Recognition Precision - gives the number of correctly identified repairs as a percentage of the number of repairs identified. This takes the number of incorrectly identified repairs into account. If the system found 50 repairs and 40 were actual repairs then the recognition precision rate would be 80%.

Correction Recall - gives the number of actual repairs corrected correctly as a percentage of the total number of repairs in the text. Therefore, if there are 50 repairs in the text and 25 of these are corrected, but only 20 of these result in the correct outcome then the correction recall rate would be 40%.

Correction Precision - gives the number of correct corrections as a percentage of the total corrections made. Therefore, if there are 40 corrections made and only 25 of these were actual repairs, but only 20 resulted in the correct outcome then the correction precision rate would be 50%.

So that:

$$\begin{aligned}
number\ of\ repairs\ in\ the\ original\ transcription &= tr \\
number\ of\ repairs\ correctly\ identified &= ci \\
total\ number\ of\ repairs\ identified &= ir \\
number\ of\ valid\ corrections\ made &= vc \\
\%\ Recognition\ Recall &= 100 \cdot \frac{ci}{tr} \qquad (1.4) \\
\%\ Recognition\ Precision &= 100 \cdot \frac{ci}{ir} \qquad (1.5) \\
\%\ Correction\ Recall &= 100 \cdot \frac{vc}{tr} \qquad (1.6) \\
\%\ Correction\ Precision &= 100 \cdot \frac{vc}{ir} \qquad (1.7)
\end{aligned}$$

This scheme is more accurate for a disfluency process than conventional accuracy measurements, used in speech recognition, as it distinguishes the disfluency process from the recognition process. Thus any errors made by the system during the normal recognition process are not shown in the results. These recall and precision values are used to measure the performance of disfluency processes that work at the sentence level. Here the string is already known and disfluencies only need to be identified and corrected within a single string. Within the recognition process the string is unknown and any disfluency process would need to work at either the acoustics, word lattice or sentences hypothesis level. Using the recall and precision measurement technique on a list of 2000 hypotheses would not be too beneficial. It is true that this technique can be used on the final output from a system but this is not accurate enough for our requirements. We need to look at whether the repair process actually gives the system the opportunity to use a corrected version. Even if the system does not use the repaired section the corrected version should be made available to the system. So rather than concentrate solely on the accuracy of the final output we will also look at the repair process' production of possible corrections and the availability of these corrections to the system. This will be done by measurements on the developing hypothesis list throughout the whole run of the system.

Therefore, a two fold analysis will be carried out. The first uses a "white box" approach by looking at internal results of the system and the progress of the repair procedure. The position of the required output (hypothesis) within the hypothesis list will be taken into account during these measurements. The second approach uses a "black box" mechanism taking the overall result into account and looking at the overall performance of the repair procedure and system.

1.4 Thesis Format

The thesis is organised according to the following plan.

Chapter 1 shows where in the field of computer science the work presented in this thesis fits. This is followed by a summary of the method used to complete the project and a discussion on the criteria for the success of the research.

Chapter 2 introduces the problem being addressed in this thesis. The general problem of human-machine communication is discussed before the problems of dealing with spontaneous speech are introduced and the specific problem of speech repair is discussed.

Chapter 3 discusses, in some detail, the current research into speech disfluencies. The approaches taken by linguists and computer scientist are compared, and a detailed survey of the leading research into disfluencies is presented.

Chapter 4 gives the results of a detailed analysis of the current system (AURAID) being used within this research. The base system, to be used throughout the rest of the research presented in this thesis, is defined.

Chapter 5 gives the results of an investigation into the performance of the AURAID system with respect to disfluent speech. It is concluded that the system has problems dealing with disfluent speech and that a process for dealing specifically with speech repairs is necessary for the system to improve and expand.

Chapter 6 investigates the speech resources that are available to speech researchers. Different forms of corpus analysis are compared and the most influential

corpora that are available are introduced.

Chapter 7 gives the results of a detailed analysis on data collected for this research. Two separate analyses are discussed and compared and a theory is introduced for dealing with speech repairs. A further discussion takes place on the different theories introduced in chapter 3 and results are discussed on analyses of these theories carried out on the data collected for this research.

Chapter 8 outlines a solution to the problem of speech repairs. A description of the general solution is given before a more detailed discussion of the solution is presented. This detailed solution deals with speech repairs at a word lattice level using the grammatical structures of repairs identified in the analysis (see chapter 7) along with knowledge on cue phrases and word fragments (collated using a sub-word pattern matching technique). The various stages to the solution are presented and the modifications to the current system are explained in detail.

Chapter 9 presents the different evaluations carried out on the final system by giving details of the data used within the analyses and various "white box" and "black box" results collated during the evaluations.

Chapter 10 concludes the thesis by discussing the overall success of the research, the advantages gained within the field of speech recognition from the research and possible future work that can be carried out with respect to the research presented in this thesis.

Chapter 2

The Problem

This chapter introduces the problem of communicating to a hard of hearing person within a lecturing environment or a group meeting. Human-machine communication is identified as a potential solution and the use of speech to communicate with a machine is discussed along with the problems that this poses. The current state of automatic speech recognition systems is discussed before the specific problem of this thesis, repair, is identified and discussed as a major problem that needs to be addressed. Speech repairs are then explained in more detail.

2.1 General Problem

The general problem being addressed by work at the University of Durham is to aid deaf people in a lecturing environment. Communication within a lecture is unique in a number of ways [Goffman and Hymes, 1981]. Primarily there is only one speaker, though the audience may participate from time to time. The topic is known to all participants and the speaker is knowledgeable in the topic and experienced in talking about the topic.

For communication to take place within a lecture it is necessary for the speaker to not only talk to the audience, but also to use displaying techniques such as blackboards, white-boards, OHP slides, etc. It may also be necessary for the lec-

turer to use some prepared notes as a reminder as to what they need to say. All of these extra activities cause problems to a hard of hearing person within a lecturing environment.

The conventional communication methods used by hard of hearing people, such as hearing aids, sign language and lip reading all present problems to both the speaker and audience. Therefore, another form of communication is necessary.

Communication could be via an interpreter who could take some of the responsibility away from the speaker to allow them to get on with the teaching. This requires an extra, fully trained, person to be employed for every lecture and would be too expensive for the majority of educational establishments. One solution would be to automate the interpretation process.

2.2 Human-Machine Communication

The problem then moves onto the communication between a "Human" and "Machine" and how to transfer what the speaker says into the machine. Human-machine communication is, at present, dominated by typing, but this is typically a very slow form of communication (although a fully trained typist can reach 100–150 words per minute). This is also a very expensive process as a fully trained typist would be involved. Handwriting is another possible form of communication which is a more universal skill than typing, but is a very slow means of communication with speeds of only about 25 words per minute. Speaking is the most common form of human communication. Speaking rates vary from about 120 to 250 words per minute making this potentially the fastest form of human-machine communication. Speech is easily learned as a child and is the most natural form of human communication. A lecturer would not require extra training and the necessity to have someone else in the room would be eliminated as the lecturer could perform the task of speaking to the system without any modifications to their current presentation technique.

ADVANTAGES	DISADVANTAGES
• Most natural form of communication between people — familiar, convenient and spontaneous. • Requires no training — people can speak, but not in all cases can they type or write efficiently. • Humans highest capacity output channel. • Allows simultaneous methods of communication — hands and voice, for example. • Allows simultaneous communication to humans and machines. • Possible in darkness, around obstacles and for the blind or handicapped. • Permits the verification of a speaker's identity. • Requires no panel space, displays or complex apparatus. • Possible at a distance and at various orientations. • Permits simultaneous use of hands and eyes for other tasks. • Permits telephone to serve as a computer terminal.	• Natural, yet unrecognisable sentences may be spoken. • Need to constrain utterances to those recognisable by machine — dependent on the application. • Speaking rate is slowed down by pauses or unfamiliarity. • Could confuse computer by speaking something to another human. • Lack of privacy if other humans are present. • Sensitive to dialects and differences in pronunciation. • Interfering "noise" can make accurate recognition difficult. • Microphone must be worn or held (closely to avoid "noise").

Figure 2.1: The Advantages and Disadvantages of Using Speech Recognition for Human-Machine Communication

Speech, therefore, is potentially the best way for a human to communicate to a machine and a visual display could be used to communicate to a hard of hearing person. Though speech is the most desirable form of human-machine communication, there are several problems in using speech as a communication method. A summary of the advantages and disadvantages of human-machine communication by speech recognition are outlined in figure 2.1 [Lea, 1980, Page 5].

There can be no doubt that automatic speech recognition is one of the most difficult "human-impersonation" tasks demanded of a computer. The ideal scenario is of any person, talking about anything, into an unobtrusive microphone, under any conditions (for example over the telephone, at an airport or with a cold), and

having their exact words or meaning immediately recognised. What happens after this step is a further problem, but in our case (for deaf students) would include a visual and/or printed reproduction.

2.3 The Current State of Automatic Speech Recognition

Early research was carried out into speech recognition, under the ARPA initiative [Klatt, 1977], but practical systems were not developed until 1983. In November 1983 the General Electric Co. introduced automatic speech recognition into its process for inspecting printed circuit boards. This was quickly followed by systems in other areas of industry along with systems for more general purposes, such as a navigation aid to a blind sailor (see Baker (1987) for more details). These systems were very limited in their use due to the constraints imposed upon them. Most of the early systems had a very limited vocabulary, were speaker dependent and only allowed the user to speak using isolated words. Automatic speech recognition has moved on rapidly since this start, though constrained requirements and technology limitations still restrict the uses of current automatic speech recognition systems.

By decreasing the complexity of the required system (i.e. limiting the domain, limiting the vocabulary and dealing with read speech alone) the solution requirement also decreases, allowing a viable solution to be produced. This solution, though adequate for the task in hand, does not normally transfer onto more complex problems. Systems that deal with read/controlled speech currently have acceptable performances, but a large amount of effort is required before little gain can be made to the performance of such systems. Moving a particular system to a new domain or onto a new style of speech requires a large amount of data, both for training acoustic models and language models. This is clearly impractical for university lectures where the topic changes constantly. To help overcome some of the current problems of automatic speech recognition systems the interest of speech recognition research has been shifting from read speech to natural/spontaneous

speech. Simply replacing read speech with spontaneous speech in a system produced for dealing with read speech can result in greatly decreased performances [Butzberger *et al.*, 1992] therefore, more detailed research is required.

The current "state of the art" can be described as large vocabulary speech recognition systems producing acceptable levels of recognition of read speech from a specific domain. Research into the move to other domains and onto spontaneous natural speech is ongoing. It is here at the cross over between read and spontaneous speech that research into automatic speech recognition systems currently lies.

2.4 The Specific Problem

There are a range of problems with using the speech of a lecturer to communicate with a machine. These problems stem from the style of the speech. Spontaneous, unrestricted speech is different from read or prepared speech in a number of ways. One of these problems is the existence of speech disfluencies. Disfluencies are the irregularities of speech which appear in spontaneous speech but are infrequent in prepared, read speech. Hindle (1983) states:

> *The essential idea is that non-fluencies occur when a speaker corrects something that he or she has already said.*

Disfluencies have been given many names including: hesitations, self-repairs and false starts. Various acoustic features have also been categorised as disfluencies (e.g. filled pauses) and different researchers have formed their own meaning of disfluencies to suit their research. The fact that standard definitions have not been formed shows that research into speech disfluencies, within the speech recognition process, is in its early stages. For this work we distinguish between three different forms of disfluency.

2.4.1 Repairs

> *A speech repair is a disfluency where some of the words that the speaker utters need to be removed in order to correctly determine the speakers meaning.*
>
> [Heeman, 1994, Page 2]

In other words, a repair is where the speaker has noticed that they have made a possible mistake during their speech, stopped speaking and either repeated what they had originally said, having decided that there was no mistake, or corrected what they had originally said by repeating the speech with some changes. The level and depth of the changes made depend on the nature of the mistake and the amount of information needed to correct the mistake.

In an analysis carried out for the research in this thesis (see chapter 7), three different repair types were identified. The first deals with repairs containing duplicate sections. Here the *reparandum*[1] matches the *correction*, though the *reparandum* could include a word fragment, such as:

> **The essen- | the essence of this course is ...**[2]

The second deals with a grammatical change to the speech. This is where a word is inserted, deleted or corrected to change the grammar of the sentence without changing the meaning, such as:

> **The first thing to tell you is the book | the recommended book for this course.**

The third deals with a semantic change to the speech. This is where a word is inserted, deleted or corrected to change the actual meaning of what is being said, such as:

> **This book | this course is not about the research aspects.**

[1] See section 2.5 for terminology explanations.

[2] The examples used in this introduction are taken from real speech.

It is this type of disfluency that the work presented in this thesis examines in detail (a fuller discussion of this type of disfluency can be found in section 2.5).

2.4.2 Filled Pauses

In certain work a filled pause (Heeman (1994) calls these abridged repairs), such as "erm" or "uh", is classed as a type of repair, while in others their presence is ignored completely. In this work a filled pause is seen as something other than a repair. It is seen as a "pause for thought" where this pause is filled with noise to stop the listeners from interrupting the speaker. The quotations above indicate that a disfluency "*corrects something that the speaker has already said*" and a repair is "*where some of the words that the speaker utters need to be removed*". It is true that filled pauses can be helpful in that they can appear in repairs and therefore their presence could indicate a possible repair. In the analyses carried out for this research, filled pauses were found both within repairs and outside repairs: 34% of filled pauses were found inside repairs, while 66% were found outside repairs. 31% of sentences contain repairs (see chapter 7, page 133) therefore the filled pauses are spread proportionally between the normal speech and repaired speech. A filled pause alone does not help in identifying the location a repair.

Shriberg (1994) claims that a filled pause is a disfluency, but says that other discourse markers categorised as fillers, such as "well" and "like", are not. Our argument is that a filled pause is a lexical entry that is used in spontaneous speech much like an interjection. Shriberg claims that different filled pauses have different meanings. If this is so then a filled pause is a "word" used in speech, not a speech repair. This is supported by the fact that filled pauses have distinct acoustic patterns [Shriberg, 1994]. If filled pauses do have distinct acoustic patterns it should be possible to recognise them and therefore deal with them quite easily. Filled pauses are more of a recognition problem at the acoustic level than a speech problem at the word level.

2.4.3 False Starts

A false start or fresh start is where a sentence was started and then ended, before it was completed and a second sentence begun. Some researchers include false starts as repairs while others treat them separately. Here we class false starts as separate from repairs because of their different nature. It is the need for part of the removed text of false starts to be kept, which distinguishes them from normal repairs, and therefore requires a different solution to that of normal repairs. We include false starts in all of the analyses carried out in this research for completeness, even though we are not specifically looking for a solution to this form of disfluency. Our analyses showed that there are four types of false starts:

1. Un-required information

   The details of the aborted sentence are not required within the dialogue and can therefore be ignored.

   Example:

   > **Its a very simple tale about | if I take these off too quickly you should shout out because I tend not to notice when you're still writing. This is a famous set of drawings.**

2. Required information

   The information from the part sentence is required to be able to understand latter (probably the following) sentences.

   Example:

   > **They are usually very difficult to maintain. When we write the program | all the ones you have written have been thrown away. In the computer industry people write programs and these programs change and evolve over the years.**

3. Major repair

   This is where it is possible, with major modifications to add (join) the part sentence to the following sentence/passage.

   Example:

   > **What they'll actually deliver is a real | I call this a whiz kid production. He spends twenty eight hours a day hunched over his machine. Putting all the bells and whistles on what is a very simple concept.**

4. Completed sentence

   This is where it is possible to add (join) the part sentence to a latter (not the following) sentence/passage. This could be seen as sentence divergence.

   Example:

   > **One interesting way of representing all of these lines of code | if I can find a piece of coloured chalk | was devised by a friend of mine.**

The problem of classification of speech disfluencies is appearing as different researchers work independently on disfluency. Here we have classified disfluencies into three different groups and will concentrate on the first group (repairs) in the rest of the work presented in this thesis.

2.5 Speech Repairs

The best way to visualise a speech repair is to look at the different definitions used to classify the sections of a repair. There are two different definitions to the "make up" of a repair, which are commonly used.

The first is based on the definition given in [Levelt, 1983] and built on in [Carletta *et al.*, 1993]. The structure is arranged around an *interruption point*. The *original utterance* precedes the *interruption point* that could be followed by

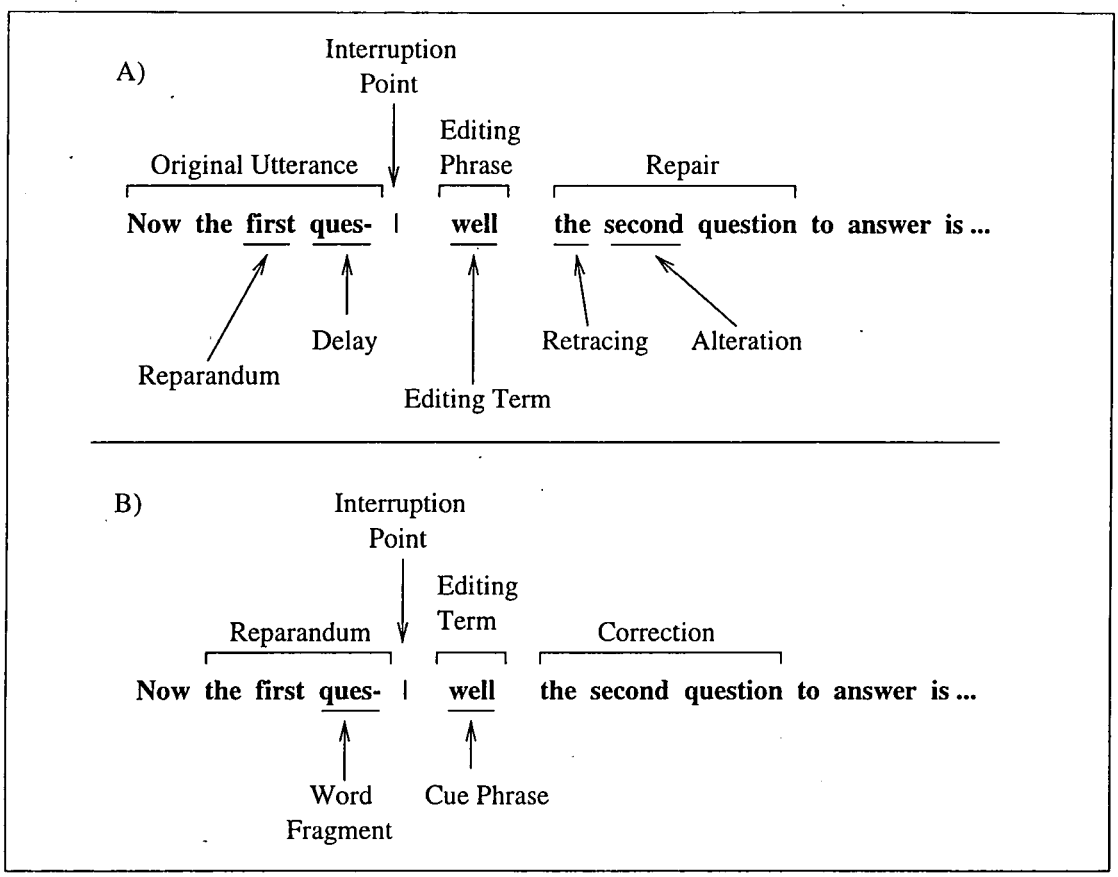


Figure 2.2: Two Repair Definitions

an *editing phrase*. The *editing phrase* may consist of a period of hesitation (pause) and/or an *editing term* such as "I mean", "sorry", "if you like", "well" or "no". The final section is the *repair* itself. Within the *original utterance* is the *reparandum* which is that piece of the *original utterance* that is not in the *repair*. The *alteration* is that piece of the *repair* that replaces the *reparandum*. There could also be a *delay* between the *reparandum* and the *interruption point* and also some *retracing* between the *interruption point* or *editing phrase*, if one exists, and the *alteration*. See section A of figure 2.2.

The second definition splits a repair into three sections: the *reparandum*, the *editing term*, and the *correction*. The *reparandum*, which might end in a *word fragment*, is the text that the speaker intends to replace. The end of the *reparandum* is called the *interruption point*, which will be marked in all examples with "|". This may then be followed by an *editing term*, which can be either a filled pause,

such as "erm" or "uh", or a *cue phrase*, such as "I mean", "sorry", "if you like", "well" or "no". The final section is the *correction* which is intended to replace the *reparandum*. In order to correct a repair, the *reparandum* and *editing term* need to be deleted. This will determine what the speaker intended to say. This gives the repair structure shown in section B of figure 2.2.

For the work presented in this thesis we use the second of these two definitions for the simple reason that it holds enough definitions for the requirements of this research. Furthermore, it conveys more closely the structure of repairs as we see them. One main advantage of the second definition is that the start of the *reparandum* is identified, which is not always true in the first definition and is important in dealing with repairs.

As can be seen from the examples given, so far, and as researchers into speech recognition systems [Butzberger *et al.*, 1992] have shown, repairs are a phenomena which are not part of written text and therefore have not been taken into account when read speech has been used within the field of speech recognition. To deal with spontaneous speech, repairs need to be taken into consideration.

2.6 Chapter Summary

This chapter began with a discussion as to why communication with a machine would be beneficial to hard of hearing people. If an automatic speech recognition system could recognise what a lecturer said and display this on a screen it would allow the hard of hearing person to undertake a normal university course, without restrictions. The lecturer should not be restricted in what they can say, nor in how they can say it. This leads to the requirement of dealing with spontaneous speech; thus introducing problems for speech recognition systems. One of these problems is the existence of various forms of speech disfluencies. The specific disfluency of repair is the one of most interest to us and will be investigated in the rest of this thesis.

The problem being tackled by the work presented in this thesis is to incorporate

into the sub-section of an automatic speech recognition system (AURAID [Collingham, 1994]), being developed for the general problem of recognising spontaneous speech within a lecturing environment, a module to deal with speech repairs, using knowledge readily available to the system and other automatic speech recognition systems.

Chapter 3

Speech Disfluencies

This chapter deals with the current state of research into speech disfluencies. A general discussion of disfluency research is given before a detailed description of the major research theories and results is presented.

3.1 Introduction

The advancement of automatic speech recognition systems over the past few years, from the recognition of isolated words, to recognising continuous speech, and now onto natural, spontaneous speech has led to a number of specific problems. In the move from isolated speech (word recognition) to continuous read speech the extra requirements of a speech recognition system included identifying word boundaries and sentence boundaries. The move from read speech to spontaneous speech has led to further problems with the grammatical nature of the speech. Spontaneous speech is known to have what is normally classed as an ungrammatical nature. Speech repairs, repetitions and false starts (disfluencies) are something that are common in spontaneous speech, but do not appear regularly in other forms of speech data, such as isolated words or even read speech. It is believed that disfluencies do not follow the normal constructs of natural speech [Hindle, 1983], therefore any passage containing a disfluency would not be recognised by an automatic speech recognition system designed to deal with grammatical speech, without some form of disfluency

processing.

Speech disfluencies are a major constituent of spontaneous speech and play an important role in the confusion of automatic speech recognition systems built for read speech. Butzberger *et al.* (1992) show that using a speech recognition system designed for read speech on spontaneous speech can "*significantly degrade recognition performance*".

Speech disfluencies are a specific problem which has arisen within the field of automatic speech recognition since spontaneous speech has become the medium of the research. Research into the structure and features of spontaneous speech is not new and has been carried out in the linguistics field for a number years [Goldman-Eisler, 1961] [Siegel and Martin, 1967] [Lalljee and Cook, 1969] [Broen and Siegel, 1972]. The problem is that this research does not tie exactly with the requirements of speech recognition research. In their analyses of the structure and features of spontaneous speech, linguists used to remove the impurities of speech, including speech disfluencies, before analysing the remaining text as can be seen from the following quote.

> *The absence of repair from the sentences with which linguists concern themselves sometimes inclines me to share the suspicion that much of the available analysis is for written sentences or for "might-as-well-be-written" sentences.*
>
> [Schegloff, 1979]

Despite this there has been some work on speech disfluencies from the linguistics field that has helped the speech recognition field [Levelt, 1983] [Levelt and Cutler, 1983] [Sagawa *et al.*, 1994].

The speech recognition field is interested in both the purities and impurities of speech, as both are an integral part of speech and both need to be handled by any automatic 'spontaneous speech' recognition system. This interest can be seen by the increased research in detecting and correcting speech disfluencies in natural speech [Hindle, 1983] [Bear *et al.*, 1992] [Hirschberg and Nakatani, 1993] [Johnson *et al.*, 1994a] [O'Shaughnessy, 1994] [Heeman and Allen, 1994a].

Due to the effect of disfluencies in spontaneous speech a system not taking account of disfluencies would in effect fail to identify 31%[1] of all possible sentences before any processing is carried out. If the system was 100% successful with all the remaining sentences the system would still only have the potential to identify 69% of the sentences correctly. This does not take into account any knock-on effects of the disfluency information or the use of semantic and pragmatic processing which are likely to use previous, incorrectly recognised, passages. It is likely that the performance would be much lower with the disfluency confusing the automatic speech recognition system. Recognition rates of this level are unacceptable and this has led to a body of research on detecting and correcting speech disfluencies. If disfluencies are added into the system as normal passages of speech (i.e. into the grammar) it may help recognition but, it does not help to identify or correct the disfluencies and would simply pass the problem onto latter processes (See section 8.1 page 153 for a discussion on statistical language models).

3.2 Linguistic Analyses

Some linguists believe that speech disfluencies are an unnatural and un-required part of speech and have thus removed them from their data. This was by no means a universal practice and there have been a number of linguistic analyses of speech disfluencies [Goldman-Eisler, 1961] [Siegel and Martin, 1967] [Lalljee and Cook, 1969] [Broen and Siegel, 1972] [Levelt and Cutler, 1983] [Levelt, 1983] [Sagawa *et al.*, 1994].

Speech disfluencies have been examined for many purposes in the fields of linguistics and psychology. Blackmer & Mitton (1991), Levelt (1983, 1989) and Sagawa *et al.* (1994) have used the information gathered on disfluencies in trying to identify how humans monitor their speech. Though not completely relevant to the requirements of speech recognition research the findings of Blackmer & Mitton and Levelt could be used within the speech recognition field. Blackmer &

[1] An analysis of speech disfluencies in chapter 7 shows that 31% of all sentences contain disfluencies.

Mitton performed latency measurements around the disfluencies found in human-human radio conversations to form a monitoring theory. This work was expanded by Sagawa *et al.* who modified the monitoring theory based on their own latency measurements. Levelt's findings have proved very useful in the field of speech recognition. Though he was investigating the human process of speech production (using speech describing coloured dot patterns arranged rectilinearly) some of his theories have been incorporated into automatic speech recognition and understanding. His "*well-formedness*" rule of speech disfluencies has been used by Heeman (1994) and Heeman & Allen (1994a) in detecting and correcting speech disfluencies. Levelt (1983) also identified that:

> *...the editing term plus the first word of the repair proper almost always contain sufficient information.*

for detecting and correcting repairs which is exactly what Hindle (1983) uses in his automatic disfluency correction process.

Levelt & Cutler (1983) examined the use of acoustic knowledge to help identify speech disfluencies. They used the pitch, amplitude and duration as prosodic markers of specific words within a disfluency. They found that phonetic repairs or word repetitions were not marked at all and only 45% of lexical repairs were actually marked. This is similar to the 38% found in earlier work [Cutler, 1983].

Though they state that prosodic knowledge is useful, one thing to come from this work is the fact that prosody is difficult to measure. Both authors analysed four hundred and twelve repairs individually and could not agree on how to solve 27% of the cases. The fact that there were limitations on the types of prosodic markers and words being analysed shows that the use of prosody is very difficult and potentially inaccurate. Nevertheless prosodic information has proved very popular in identifying disfluencies in both linguistics [Howell and Young, 1991] [Beach, 1991] and speech recognition [Nakatani and Hirschberg, 1994].

Howell & Young (1991) add extra prosodic knowledge to that identified by Levelt & Cutler (1983). In their analysis of three hundred and ninety one speech

repairs taken from the LLC[2] they found that a pause directly after the *reparandum* and increased stress on the first word of the "alteration"[3] compared to that of the replaced word in the *reparandum* are important. In perceptual experiments they found that listeners could comprehend those disfluencies that contained prosodic information and that the listeners could repeat the repaired speech quicker if, both pause and increased stress is available. Like most work in speech disfluencies the authors removed one hundred and nineteen (30%) of the disfluencies from the analysis for various reasons, such as, overlapping disfluencies, the existence of *cue phrases*, disagreement about the true prosodic marks and multiple interpretations.

Beach (1991) believes that prosody plays an important role in speech production and recognition and claims that "*there is a relationship between sentence prosody and syntactic structure*". This is a notion agreed with by Price *et al.* (1991), who go on to examine the role of prosody in distinguishing the meaning of an otherwise ambiguous structure. If prosody is actually used in this way then it is possible to believe that prosody does play an important role in distinguishing the role of speech disfluencies and the actual meaning of an ambiguous passage containing disfluencies. Speech recognition researchers have picked up on this and much interest has been shown in the use of prosody within speech recognition systems [Bear *et al.*, 1992] [Nakatani and Hirschberg, 1994].

From a psycholinguistic point of view (i.e. how the human process works rather than analysing the disfluencies themselves from a computational point of view), one analysis of a large number of speech disfluencies has been carried out by Shriberg (1994). Shriberg attempts to form a stable platform from which disfluency researchers can build. The goal of the work was to provide evidence that disfluencies, within spontaneous speech, show regular trends in a number of dimensions. A database of 5,025 speech disfluencies was produced, taken from three speech corpora of dialogues, and marked using knowledge from various sources including: speaker information, sentence information, and a pattern labelling scheme based

[2]The London-Lund Corpus, see chapter 6 page 119.

[3]The alteration starts at the word that replaces the incorrect word in the *reparandum* (using the notation used in this thesis) and is not the *correction*, as defined earlier in this thesis.

on that of Bear *et al.* (1993) . This pattern labelling scheme was used to group the disfluencies into eight different types, which were then used in the analysis to label the different forms of disfluencies. A large number of facts on the disfluencies and various trends were noted. Though "syntactic features" were not used within the data, it does show that:

> *...there are, indeed, significant regularities in the distribution and characteristics of disfluencies.*
>
> [Shriberg, 1994]

This is something which is required if automatic speech recognition systems are to deal with speech disfluencies.

Work at Edinburgh university has examined how points of disfluency are recognised within the human recognition process [Lickley and Bard, 1992] [Lickley, 1994]. They investigated the possibility that prosody is not the fundamental factor in disfluency identification and hypothesised that syntax also plays an important role. They performed perception tests of disfluencies by stopping the text before and during the onset of the continuation word (i.e. the word following the *interruption point*). This showed that it was possible to detect the disfluency before the continuation word was complete, and recognised. Thus indicating that it is unlikely that syntax plays a major role in the human perception of speech disfluencies. They also noted that the disfluency was generally recognised after the onset of the continuation word, but very rarely before the onset of the continuation word. This points to the fact that there is a major source of information at the onset of the continuation word, but no hypothesis was given as to what this could be (though it was suggested that the expectancy or rhythm of the prosodic/acoustic signal could play a part).

Oviatt (1995) has investigated the design of a user interface by comparing disfluencies found in a number of different communication media (writing, human-human communication and human-computer communication). The domain was information extraction and the goal was to provide empirical guidance for the design of

robust spoken language technology. Oviatt found that two factors influenced the disfluency rates.

- the length of the utterance.
- the lack of structure within the presentation format.

Oviatt hypothesised that design methods capable of guiding users' speech into briefer sentences have the potential to eliminate the majority of spoken disfluencies. In her analysis, Oviatt found that "*a structured presentation format successfully eliminated 60-70% of all disfluent speech*". This is not too helpful for the requirements of a lecturing environment, but it does show that prevention rather than the cure of speech disfluencies is being investigated.

Fox Tree (1995) investigated the effect of disfluencies on understanding, and in-particular the understanding of the words following a disfluency. The findings were that false starts cause greater problems than repetitions. Fox Tree also looked at current theories of speech production and compared the results and the theories.

Fink & Biermann (1986) investigated the use of expectations within a dialogue to anticipate the next input. Using information on previous dialogue structures and the current trends of the speaker, taking note of repetitions in the dialogue, the system gives the expected meaning to a parser. The parser is then strongly biased towards this meaning. A system was developed and used to correct miss-recognitions by a speech recognition system. The theory could easily be moved into an automatic speech recognition system, though it would only really work in a structured dialogue situation.

3.3 Speech Recognition Analyses

Although there was early research [Goldman-Eisler, 1961] [Lalljee and Cook, 1969] [Broen and Siegel, 1972] into the process of speech disfluencies, it was not until 1983 [Hindle, 1983] that the first attempt to automate the process of correcting

speech disfluencies was performed. Only recently has there been an attempt to expand the theories of dealing with speech disfluencies so that they can be used within automatic speech recognition systems. A number of theories have built upon the work of linguists and incorporated this into speech recognition research. Carletta *et al.* (1993) have developed a coding scheme for repairs similar to that used in [Levelt, 1989] and have analysed ten dialogues from the Map Task Corpus using this coding scheme. Heeman (1994) has used Levelts "well-formedness" theory and Hindle (1983), though not from the speech recognition field (he does produce an automatic method to deal with speech disfluencies), has also used theories similar to those of Levelt. A further source of information suggested by linguists is the use of acoustic/prosodic information.

The views of psychologists and linguists is that prosodic features play an important role in speech disfluencies. The problem is that little is known as to the exact nature of prosody and prosodic marks. Speech recognition researchers have tried to identify exactly what the different prosodic marks actually mean within sentences and across sentences but as yet no complete theory exists. Work is continuing and as more information is gathered about prosodic marks the true nature of prosody within spontaneous speech may well become clear.

Prosodic information has also been shown to have an ambiguous nature and can therefore not be used alone. The extra information required could include pattern matching techniques or simple syntactic information. Both have been investigated as possible approaches to identify and correct speech disfluencies.

Two distinct approaches to dealing with speech disfluencies have appeared within speech recognition research. The first uses acoustic knowledge to identify a disfluency and the extent of the disfluency, while the second uses pattern matching techniques and limited syntactic knowledge. Only rarely are the two combined into a unified theory [Bear *et al.*, 1992]. Though many will agree that this is the ultimate solution there is currently insufficient knowledge about either aspect for a unified theory.

As interest in this area is increasing more research is being carried out. Below

is a list of the different people and sites that have contributed to the ever growing research into speech disfluencies.

3.3.1 INRS (Quebec)

One of the main pieces of work on using acoustics within the speech recognition field is given in [O'Shaughnessy, 1992] [O'Shaughnessy, 1993a] [O'Shaughnessy, 1993b] [O'Shaughnessy, 1994]. O'Shaughnessy examined the use of acoustic knowledge on pause lengths, word durations and fundamental frequency (pitch contour) in the identification of, what he calls, false starts (normally classed as disfluencies) within spontaneous speech.

The data was taken from the ATIS domain. The passages consisted of forty two adult male and female speakers each speaking thirty utterances. The utterance lengths varied with an average utterance length of twelve words. Therefore 1,260 utterances consisting of approximately 15,120 words were analysed. From these utterances sixty word repetitions, thirty word insertions and twenty five word replacements were found. In all, approximately 10% of the utterances contained speech disfluencies.

The early analyses [O'Shaughnessy, 1992] [O'Shaughnessy, 1993a] [O'Shaughnessy, 1993b] identified that there were prosodic factors that could be used for the identification of "simple" restarts (i.e. repetitions) and used a basic rule of:

> *If a pause is less than 400ms then there is a false start.*
>
> [O'Shaughnessy, 1992]

This rule gave correct identification of 70% of restarts and only 35% of the selected locations were false identifications, giving a recognition recall[4] rate of 70% and a precision rate of 65%. This performance is well above chance but it was clear that pause duration alone was not a reliable cue of simple restarts. The theory was expanded so that simple restarts could be corrected. It was noted

[4]See section 1.3, page 6, for a definition of these measurements.

that the acoustic patterns of the repeated words changed very little and therefore an analysis of the "spectral-time patterns" before and after the *interruption point* (relevant pause) could be carried out. O'Shaughnessy recommended that only two to three syllables need be examined. By comparing the three syllables before the identified *interruption point* and the three syllables after the *interruption point* it was expected that the first of two 'matching' spectral patterns could be removed, thus correcting the repair. No results of this are given but it was hypothesised that this process would increase the identification of repairs to 80% and reduce the level of false alarms to 15% giving a recall rate of 80% and a precision rate of 85%.

To deal with more complex restarts, which contain insertions and substitutions, it would be necessary to include a more detailed analysis of the surrounding syllables. But more interesting is the fact that the author states:

> *... there are many possibilities here and many of them have spectral and prosodic patterns that resemble fluent speech.*
>
> [O'Shaughnessy, 1992]

So in his analysis O'Shaughnessy has found that prosody has an ambiguous nature and therefore to analyse a specific problem with respect to prosody alone seems to be inadequate.

As an extension to his earlier work O'Shaughnessy examined complex restarts [O'Shaughnessy, 1994], which are those defined as involving the deletion and/or insertion of word(s). The theory which was developed was based on the original 1,260 utterances and was tested on a separate set of 2,132 utterances from the ATIS corpus. These 2,132 utterances contained thirty two simple patterns, forty complex restarts with matching words and twenty five complex restarts without matching words. This latter set of disfluencies were not taken into account in this analysis as:

> *... the speakers did not help the listener decode the restart by repeating words, but simply resumed speaking after a brief pause, assuming that the listener would delete the immediately preceding word(s).*
>
> [O'Shaughnessy, 1994]

His solution was to take the syllable matching technique used in the earlier work (expanding it to one, two or three syllables) and allow up to four syllables between the matching sequences. So the structure of the disfluencies was expected to be:

(1-3 matching syllables) <pause> (0-4 syllables) (1-3 matching syllables)

Using this method on the 2,132 test utterances "*preliminary results indicate successful recognition in approximately half of the complex restarts involving repeated words*". No actual figures were given on the performance of the system except the above, though he also states that "*there appeared to be a relatively low rate of false alarms*".

Of most interest to the work in this thesis is the fact that the solution attempts to deal with speech disfluencies at a sub-word level, without the identification of the words or even word boundaries. There are a number of problems with this approach. The first is the number of repairs covered by the algorithm. All of the repairs are expected to contain a matching section, which is not the case and those that do are expected to have matching sections of between one to three syllables in length. Therefore the *reparandum* can only ever be up to three syllables in length. Any newly added words after the *interruption point* can only span upto four syllables. These limitation eliminate a number of disfluencies and are too limiting to make the algorithm practical. It could be possible to increase the length of the different sections of the structure but O'Shaughnessy (1994) states:

> *...increasing the length of added new words beyond four syllables resulted in too many false alarms.*

The algorithm also does not take word fragments or cue phrases into account. Disfluencies containing word fragments are simply removed from the initial list of disfluencies that can be solved and *cue phrases* are ignored completely. A further problem with dealing with repairs at the acoustic level is that any information removed at this stage would not be available later, therefore any mistakes made here would propagate through the whole of the recognition process without the possibility of re-trace.

3.3.2 ATT & Harvard

Work into investigating acoustic and prosodic cues to disfluencies within the ATIS corpus has also been performed [Hirschberg and Nakatani, 1993] [Nakatani and Hirschberg, 1993a] [Nakatani and Hirschberg, 1993b] [Nakatani and Hirschberg, 1994]. They split their disfluencies into three intervals and investigated the acoustic properties of each interval. Using a decision tree mechanism they managed to identify one hundred and ninety two (86%) out of two hundred and twenty three speech disfluencies with only nineteen false positives (giving an identification recall rate of 86% and a precision rate of 91%). They used knowledge on pause durations, word fragments, filled pauses, amplitude comparisons, F0 changes and some lexical knowledge on word lengths and word tags. Their main source of knowledge was the duration of the pause after the *interruption point*, which has figured in a number of acoustic investigations of disfluencies, but they also used the existence of a word fragment as a major key. The existence of a word fragment was used to identify one hundred and six (55%) of the one hundred and ninety two identified disfluencies. The authors did examine word fragments in more detail, but did not come to any conclusion as to which factors could be used to identify a word fragment. The fact that word fragments and filled pauses were used heavily as identification markers and the fact that these were entered manually decreases the significance of these results.

3.3.3 Hindle

One of the earliest pieces of work on automating the detection and correction of speech disfluencies was carried out by Hindle (1983). This work included a wide range of disfluency types from repairs to false starts and was based on work by Labov (1966). Labov identified a phonetic marker within the speech signal that was said to identify where a disfluency had taken place. With this marker Labov formed a simple set of rules, based on "pattern matching" within the word string, to identify the type of disfluency and the correction required. This phonetic marker is also presumed in the work of Hindle, though the exact nature of this signal is as

yet unknown, but Hindle goes deeper into the "syntactic constituent-hood" of the disfluency and moves away from the string level pattern matching technique.

Hindle used an interview scenario to collect a corpus of twenty hours of human-human dialogue of which a ninety minute passage was used in his analysis. Of the sentences within the ninety minute passage Hindle found that 50% contained repairs. This is a very high figure but is not totally unexpected given the nature of the dialogue.

Hindle's system was built on top of a deterministic parser using a 2,000 word lexicon. A number of additional rules were added to the parser to allow it to deal with disfluencies by ignoring the *reparandum* when processing a string. Hindle classed his rules into two different groups. The first dealt with the "expunction of copies" while the second dealt with "lexically triggered restarts". These rules have been summarised into four basic rules.

1. If the words before and after an edit signal (*interruption point*) are the same then remove the first occurrence.
2. If the class of the word after the *interruption point* matches the class of the word before the *interruption point* then remove the first word.
3. If the class of any word before the *interruption point* matches the class of the word directly after the *interruption point* then remove this word and all of its joined constituents.
4. If the words after the *interruption point* are a *cue phrase* then remove everything before the *interruption point* (this was to deal with restarts).

Hindle showed some promising results and indicated that 95% of the disfluencies in his test sample were processed correctly. Only 3% resulted in a correction being made that failed to produce the required result while no correction was attempted for only 2% of the disfluencies.

One of the main problems with this work is the amount of pre-processing required for the system to work. This pre-processing includes:

- Manual indication of sentence boundaries.
- Manual indication of speaker changes.
- Identifying the position of the *interruption points.*
- Manual changes to syntactic tags.
- Manual identification of word fragments and their required tags.

A further problem with this processing technique is that the system is presumed to know exactly what has been said and only needs to identify disfluencies within this string. It does not need to identify the disfluency within an unknown string or a number of possible strings, which is the real problem within an automatic speech recognition system. The fact that word fragments are available and tags are added manually also simplifies the problem. Moreover, the 'major' drawback of this approach is the manual identification of the *interruption point* (i.e. exactly where a disfluency occurs). This removes the possibility of false positives and overcomes one of the main problems in the detection of speech disfluencies. The process of dealing with false starts is also a problem. Not all false starts contain a *cue phrase*, plus, as indicated in our analysis of speech disfluencies (see chapter 7), it is not acceptable to simply remove the *reparandum* of a false start as some information loss could result.

If a clear edit signal can be identified and the problems of identifying word fragments and tagging strings of words overcome, then this process does show some potential.

3.3.4 SRI International

Bear *et al.* (1992) and Shriberg *et al.* (1992) have investigated the automatic identification and correction of repairs, but do not take into account the problem of

|	- The *interruption point.*
M	- Matching words
R	- Replacements
^	- Joined words
X	- Insertions and Deletions
C	- Cue words
FP	- Filled Pauses
-	- Fragment marker.

Figure 3.1: Pattern Matching Notations used by Bear *et al.* (1993)

false starts. Three stages and three forms of knowledge are used in an attempt to solve repairs. The first stage uses a simple pattern matching technique to identify repeated words, which are common in repairs. They then incorporate parsing techniques to clarify the existence of a repair and finally incorporate acoustic/prosodic information to help identify the type of the repair.

The data they used was again from the ATIS corpus of human-machine dialogues, where the speech was recorded using a "wizard of oz" technique, within the field of flight bookings. From their 10,718 sentences they found six hundred and seven (5.6%) contained speech repairs and that 10% of those sentences of more than nine words contained repairs. This is very low but it is likely to be due to the fact that the nature of the speech leads to a very structured conversation.

Their solution is based on three phases:

i) Pattern Matching

The first phase uses a pattern matching component to search for identical sequences of words and simple syntactic anomalies such as "a the" or "to from". The pattern matching notation used within this work is given in [Bear *et al.*, 1993] and a summary is given in figure 3.1.

Some examples of repairs and their patterns are:

```
List the aircraft | list the types of aircraft that ...
M1  M2    M3    | M1 M2   X   X   M3

On July fif-  | on July twentieth ...
M1 M2  R1-  | M1 M2    R1

back to Pittsburgh I'm sorry | back to Denver
M1  M2    R1        C    C   |  M1 M2   R1
```

It should be noted that in this notation the *interruption point* is after the *cue phrase.* Other work in speech disfluencies has the *interruption point* before the *cue phrase* and this latter notation is used in this thesis.

The number associated with each pattern class is used to indicate which word matches or replaces which word on the opposite side of the *interruption point.* So that in a repair, "M1" in the *correction* would match "M1" in the *reparandum* and "R2" in the *correction* would replace "R2" in the *reparandum.*

Using only this first phase a test using four hundred and six of the 'non-trivial' repairs was carried out. The system successfully found three hundred and nine (71%), but the system also identified one hundred and ninety one (38% of all those identified) false positives giving an identification recall rate of 71% and a precision rate of 62%. Of the three hundred and nine correctly identified the system managed to correct 57% giving a correction recall rate of 43% and a precision rate of 35%.

ii) Parsing

The purpose of the second phase was to eliminate the false positives identified during the first phase. The idea was to parse each of those sentences identified as containing a possible repair and remove those that parsed successfully. It was expected that an actual repair would not parse while the false positives would. A test using this theory was carried out on three hundred and thirty five sentences of which one hundred and seventy nine contained actual repairs and one hundred and seventy six contained false positives. Two tests were carried out, one using syntactic information only, and the other using sytactic and semantic information.

			Results		
	type	*total*	*repair*	*false*	*un-decided*
Syntax only	repair	179	68	56	55
	false	176	3	131	42
Syntax and Semantics	repair	179	64	23	92
	false	176	11	90	75

Table 3.1: SRI International: Parsing Results

Table 3.1 shows the results of these tests.

Though this technique does seem to eliminate the majority of the false positives it also eliminates a number of the actual repairs. These results though interesting do not seem to give an adequate medium for identifying repairs. It is true that of the sixty eight identified as a repair (using syntactic information), sixty two were actually corrected correctly and sixty of the sixty four repairs identified as speech repairs using the semantic information were also corrected correctly. This technique shows promise in identifying repairs that can be correctly dealt with, but it does eliminate a large number of repairs that should still be handled.

iii) Acoustics

The purpose of the third phase was to investigate the use of acoustic information in distinguishing false positives from repairs. An analysis was carried out on two types of repairs: those that had the pattern "M1 | M1", and those that had the pattern "M1 | X M1". It was found that there was a difference in the pause lengths and word durations between the false positives and repairs. Actual tests were not carried out and only hypotheses made based on the analysis, but it was claimed that:

> *... acoustic information can be quite effective when combined with other sources of information, in particular with pattern matching.*
>
> [Bear *et al.*, 1992]

The main problem, with the approach of SRI International, seems to be with the number of false positives and the results of the parsing phase. The parsing phase

removes a large number of the actual repairs. A problem not considered by Bear *et al.* (as far as speech recognition is concerned) is the use of these techniques with a number of hypotheses, as produced by an automatic speech recognition system. They deal with the actual strings that include word fragments etc. Identification of a repair by mapping the prosodic marks onto the text is relatively straight forward when compared to identifying the possibility of a repair given a number of hypothesised text strings and prosodic marks.

To help overcome this, pattern matching techniques have also been used within a Spoken Language Understanding system. Developed at SRI [Dowding *et al.*, 1993a] [Dowding *et al.*, 1993b], GEMINI uses a parser-first approach in dealing with speech repairs. The system first parses any input. If no semantically acceptable interpretation is found for the complete utterance then a repair correction process, based on the above pattern matching technique, is used as a fallback method. From a training passage of 5,875 utterances (containing one hundred and seventy eight repairs) eighty nine (50%) repairs were found and fifteen false positives were identified. Giving an identification recall rate of 50% and precision rate of 86%. From the eighty nine identified repairs eighty one were modified correctly giving a correction recall rate of 46% and a precision rate of 78%. From test data of seven hundred and fifty six utterances containing twenty six repairs, eleven were found along with two false positives giving an identification recall rate of 42% and a precision rate of 85%. Of the eleven repairs identified eight were modified correctly giving a correction recall rate of 31% and a precision rate of 62%.

Because of the nature of their matching system it is only possible for it to deal with those repairs that contain matching words. Any other pattern would result in a large number of false positives. Consider the repair [Bear *et al.*, 1993]:

What are the cheap | cheapest one way flights ....

and its repair pattern "R1 — R1" (i.e. when one word follows another then the second word could replace the first word). Matching this onto a sentence would result in every combination of two words being identified as a possible repair.

3.3.5 Rochester

Work carried out at Rochester University has developed a method of detecting speech repairs using a combination of two processes [Heeman, 1994] [Heeman and Allen, 1994a] [Heeman and Allen, 1994b]. The first is similar to the work by Bear *et al.* (1992) and uses a pattern matching technique while the second process uses a part-of-speech tagger to supply information to the pattern matcher and to provide a statistical opinion on the pattern matchers recommendations.

The data used in the analysis was collected for the TRAINS project [Allen *et al.*, 1994]. One hundred and twelve problem solving, human-human, dialogues were recorded. For this work the training data was taken from forty (24,000 words) of the dialogues and the test data was taken from seven (5,800 words) of the dialogues. In the test data seven hundred and twenty five speech repairs were found. Of these two hundred and sixty seven were abridged repairs (i.e. where a word fragment or *cue phrase* is the only addition to the text and no actual change to the structure was made) and four hundred and fifty were modification repairs (i.e. word repetitions, word replacements and word insertions). Eight repairs were removed from the data as it was impossible to manually distinguish them from overlapping repairs. This left seven hundred and seventeen abridged and modification repairs. Of these seven hundred and seventeen repairs 26.5% contained a word fragment and 32.4% contained a *cue phrase*.

Three approaches are taken in the Heeman process:

i) Pattern Matching

The pattern matching technique is based on the work of Bear *et al.* (1992), but Heeman claims that the pattern matcher of Bear *et al.* is too rigid and needs to be more flexible and lenient. Therefore, Heeman uses three generic patterns to cover the seventy two repair patterns found in their four hundred and fifty modification repairs. These generic patterns are:

- M | M (with upto three intervening words)

- MM | MM (with upto six intervening words)
- R | R (with zero intervening words)

The pattern "R | R" generally means replace one word with another but Heeman adds the proviso that these words must be of the same category. Therefore, the final pattern matches adjacent words with the same syntactic category. It is claimed that using these three generic rules seven hundred and eight (98%) of the seven hundred and twenty five speech repairs can be identified. There are problems in that a large number of false positives would be identified as possible repairs and the patterns themselves do not identify the extent of the repair or the final pattern of the repair.

Once the location of a possible repair is identified, using these generic rules, the next step is to identify the correction that would produce the required output. The technique used for this is a more detailed pattern matching process. The patterns used by Heeman are not identified before processing, but are built as the input text is processed, using a set of rules, with the aim of "*capturing the notion of the well-formedness of speech repairs*" (as identified in [Levelt, 1983]). These rules are as follows:

1. *Cue phrases* must be adjacent.
2. *Cue phrases* must immediately follow the *interruption point.*
3. A fragment, if present, must immediately precede the *interruption point.*
4. Word correspondences must straddle the *interruption point* and can not be marked on a word labelled as a *cue phrase* or fragment.
5. Word correspondences must be cross-serial; a word correspondence cannot be embedded inside another correspondence.
6. If there are no other word correspondences, there can only be three intervening words, excluding fragments or *cue phrase*, between the first part and the second part of the correspondence.
7. In the *reparandum*, two adjacent matches can have at most four intervening words.

8. In the *correction*, two adjacent matches can have at most four intervening words.

9. For two adjacent matches, the number of intervening words in the *reparandum* can be at most one more than the number of intervening words in the *correction.*

10. A word replacement must either only have a fragment and *cue phrase* between the two words that it marks, or there must be a word correspondence in which there are no intervening words in either the *reparandum* or the *correction.*

From the possible repair locations identified using the generic patterns a specific pattern is built using these rules. If this pattern is "well-formed" then it is possible that there is a repair. There are three criteria that point to the "well-formedness" of a repair. The first is the occurrence of a *cue phrase*, the second is the occurrence of a word fragment and the third is when there are no intervening words between the end of the *reparandum* and the start of the *correction.* If any of these criteria are met then the pattern is identified as a possible repair.

	Training Set	**Test Set**
identification recall	95%	92%
identification precision	56%	45%
correction recall	89%	86%
correction precision	52%	43%

Table 3.2: Rochester: Pattern Matching Results

The results of this combined pattern building technique are quite promising. Recall rates are high but the precision is low as there are still a number of false positives identified. The results are shown in table 3.2.

	Simple Module	Frag-ments	Cue Phrases	Word Match	Full
Training:					
Recall	44%	50%	45%	77%	79%
Precision	45%	48%	47%	55%	60%
Testing:					
Recall	31%	44%	33%	75%	77%
Precision	57%	62%	59%	58%	62%

Table 3.3: Rochester: Tagging Detection Results

ii) Part-Of-Speech Tagger

There are three reasons for using a part-of-speech tagger. The first is that the pattern matcher requires the classes of words to identify replacements. The second is that the pattern builder requires *cue phrases* and word fragments to be identified. The third reason is to attempt to identify the *interruption point* of the repairs and distinguish between modification and abridged repairs, as the author claims:

> *One powerful predictor of modification repairs is the presence of a syntactic anomaly at the interruption point.*
>
> [Heeman and Allen, 1994b]

A part-of-speech tagger takes a string of words and gives each word a syntactic class based on the classes of previous words and a statistical model of all possible class sequences. Because of the problems caused by repairs the tagger used by Heeman & Allen, is given knowledge about the category transitions for the *interruption points* of the repairs in the training data, and so it is able to mark a transition either as a likely repair or as fluent speech. Other contextual knowledge, such as *cue phrases*, word fragments and word matchings are also factored in by modifying the transition probabilities.

Table 3.3 gives the detection results using the different knowledge incorporated into the part-of-speech tagger. Of the three hundred and eighty four modification repairs three hundred and five were found using the full system and two hundred

	Training Set	Test Set
detection recall	91%	83%
detection precision	96%	89%
correction recall	88%	80%
correction precision	93%	86%

Table 3.4: Rochester: Combined Results

and seven false positives were also found. This is an increase over the one hundred and sixty nine of the original system but this is not unexpected as the manual inclusion of the word fragments and *cue phrases* identify exactly where an *interruption point* is.

iii) Combined approach

Pattern matching on its own produces too many false positives and part-of-speech tagging does not identify enough of the speech repairs, therefore, a combined approach is taken. First the words are tagged. They are then passed to the pattern builder to identify possible repair locations and structures. From these repair structures the knowledge on repair locations, produced by the tagger, is used to identify the possible existence of a repair.

Patterns containing only word repetitions are automatically classed as repairs without the extra knowledge from the tagger. The rest of the patterns are marked as modification repairs if the tagger identified an *interruption point* in the appropriate place within the repair pattern. If a repair is rejected, i.e. not marked as a modification repair, but contains a word fragment or *cue phrase* then it is classed as an abridged repair.

The results (see table 3.4) are very promising. However, it is difficult to understand why the processes were used in the way they were. If the final decision is down to the tagger then those patterns identified, using the pattern matching process, would not be identified as a repair. Therefore, the most obvious process would be to use the location identified by the tagger first and then use the pattern

building rules to span the identified *interruption point.* From the figures (two hundred and sixty seven abridged repairs and sixty six modification repairs with word fragments or *cue phrases*) we can see that three hundred and thirty three repairs would be identified using the word fragment or *cue phrase* information. This, when added to the three hundred and five repairs identified using the tagger alone, would give six hundred and thirty two repairs (88%) identified and only two hundred and seven false alarms. The pattern matching could then remove those possible repairs that don't have a well-formed structure. It seems as if this would produce the same result without the need for the three generic patterns.

Heeman & Loken-Kim (1995) have identified the fact that the statistical analysis makes the final decision as to what is a repair and what is not is a problem and have examined using information on the structure of the potential repair to modify the decision of the statistical analyser. Here the repair patterns are categorised into ten different groups, using knowledge on the matches present, amount of change made, structure of simple patterns and certain clues to the location of the *interruption point.* These ten categories are given probability factors for their likelihood of being a modification repair. The system then re-calculates the probability of the *interruption point* being for a modification repair based on the proposed pattern of the proposed repair and the category that the pattern fits into. Each category probability is used to modify the repair probability given by the syntactic anomaly alone. Using this knowledge the recall rate increased from 72% to 75% and the precision rate increased from 69% to 77%. There is still the problem that if the syntactic system does not identify the actual *interruption point* then the repair will never be identified.

Further problems with this approach are similar to the other approaches mentioned. Word fragments and *cue phrases* exist and are identified. The process works at the word level, in that the string spoken already exists:

> *... as would be provided by an ideal speech recogniser.*
>
> [Heeman and Loken-Kim, 1995]

This approach deals with the understanding of speech and the fact that speech repairs hinder understanding. As has been shown in an analysis of repairs within an automatic speech recognition system (chapter 5), one of the main problems is recognising the string in the first place. Dealing with speech repairs at the dialogue level could be too late in the process of dealing with spontaneous speech. The system also makes no use of the word fragment as a possible match which is normally the case in speech repairs. It is true that the fragment would match the end of the resumed text and would therefore not break the pattern building rules, but there is information there that could be exploited.

3.3.6 SRI Cambridge

One system which uses repair processing in a practical way is the Spoken Language Translator developed by SRI [Agnas *et al.*, 1994] [Carter and Rayner, 1994] [Rayner *et al.*, 1993] [Rayner *et al.*, 1994]. The purpose of this system is to translate spoken English into spoken Swedish. The system uses a pipeline approach to joining a speech recogniser, language processor and speech synthesiser together. It uses the DECIPHER speech recognition process to produce an N-best list of hypotheses (five is found to be the optimum level). These hypotheses are then passed onto the Core Language Engine for linguistic analysis. The outcome of this linguistic analysis is a "meaning" of what has been said. This is then used to produce a string for the Swedish synthesiser, PROPHON, to produce the actual spoken output.

The repair process is built into the linguistic analysis process. For each sentence hypothesis provided by the speech recogniser, repair processing is carried out. Repairs are identified and a copy is made of the original hypothesis and the repair corrected. The original hypothesis is kept for subsequent processing so as not to rely solely on the repair identification process. Further linguistic analysis is performed on both the repaired hypothesis and the original hypothesis.

The repair process deals with repairs at the string level and incorporates some syntactic features when required. First, a search is made for repeated grammatical roots such as "flight ... flight" or "is ... are". When a repetition is found the

possibility of a repair is noted. The following example will be used throughout the explanation of this process:

I want to go from Boston no from Denver to Boston on Tuesday.

The second stage is to remove those that are obviously not repairs by discarding those that have no intervening words and are made up of repeated roots of common words such as "to", "a", "from", "and", "in" and "or". The remaining possible repairs are then scored using a simple scoring mechanism. Two points are given to the repair if there is a matching root and one point is deducted if the process needs to jump a word to find a matching sequence. In the example "from" and "Boston" would give four points but "Denver" and "to" would give minus two points. If there are no intervening words (in our example there is an intervening word of "no") then the repair is passed forward as having the identified score. If there are intervening words then the process tries to join these to either the end of the *reparandum* or the beginning of the *correction*. It does this by trying to match "no" with "on" and "no" with "go" in our example. A match is true if both words are of the same major word class. If there is a match then the details of the repair are changed (i.e. the length of the repair sections) and the repair with its unmodified score is passed as a possible repair. If the intervening word(s) can not be matched then a check is made to see if it is a *cue phrase*. If this is true then one point is added to the repair score and the details are changed. In our example "no" would be classed as a *cue phrase* giving an overall repair score of three, a *reparandum* of "from Boston no" and a *correction* of "from Denver to Boston".

No conclusive results were given with the process description. Of 4,615 sentences taken from the ATIS corpus one hundred and thirty possible repairs were identified by the process and eighty one of these were actually used by the language system in producing the meaning of the original spoken passage. Of these eighty one, seventy seven were correct "*or as plausible as any other choice available*" and only four were incorrect. It is difficult to see what these results mean as no pre or post analysis was carried out on the data. It is therefore not possible to know how many repairs were actually in the original 4,615 sentences. Using results from other analyses

on the ATIS corpus, we would expect approximately four hundred and fifty (10%) repairs to be present in the data. This means that only 17% of the repairs were being corrected. Though the precision of the process seems to be high. The fact that the process does not cover all of the expected repairs is a problem. A further problem is the timing of the repair process. As discussed earlier, repairs cause problems for automatic speech recognition systems and using a repair process on an N-best list could be too late in the process of dealing with spontaneous speech.

3.3.7 Durham

The work carried out at Durham and presented in detail later[5] within this thesis deals with the identification and correction of speech repairs in a practical environment. Most repair procedures deal with repairs at the string level. This is too late within the recognition process as grammatical problems, posed by repairs, will be encountered before a full transcription, including repairs, could be produced. The work at Durham deals with repairs at the word lattice level. Here many possible words and their grammatical tags are present and grammatical knowledge has not limited the possible solutions. This allows repairs to be identified and corrected before any grammar is used within the system. By traversing the word lattice and identifying repair structures, corrections can be made within the lattice to allow a subsequent hypothesis generation process to ignore those words that make up the *reparandum* of the repair, thus correcting the repair.

Repair structures have been identified in an analysis (see chapter 7) of natural speech and these structures are the basis for identifying repairs within the lattice and identifying the extent of the repair and therefore, the required solution. Knowledge on potential word fragments, produced using a phoneme pattern matching process, and *cue phrases* are identified so that the repair process can automatically deal with some of the problems posed by repairs.

This is a practical solution to speech repairs and has shown good results in a

[5]Chapter 8 gives a detailed description of the solution and chapter 9 shows results of the implemented solution.

number of tests carried out on the system (see chapter 9). Limited changes are required to the hypothesis generation process and only one module needs to be added to the initial system.

3.4 Chapter Summary

It is very difficult to compare the performances of the different theories and processes developed for dealing with disfluencies. Each have their own merits and problems and use their own definitions of repairs and disfluencies. One of the main problems is that the majority of the theories produced for speech recognition, work at the string level. In our view this is too late in the process of speech recognition and understanding. O'Shaughnessy has investigated disfluency identification at a very early stage but is too limiting in his coverage of repair structures. What is required is a technique that works at a sub-sentence level, but possibly at a pre-word level. This gives the advantage of knowing, possibly, what the words are without being constrained to a single sentence interpretation.

The areas of both acoustic knowledge and syntactic knowledge for repair analysis has been investigated to some degree. Though syntactic knowledge has been limited to pattern matching and the syntax of specific words within the disfluency, it has shown the most potential. The most promising approach appears to be a combination approach between the two forms of knowledge. But as Shriberg has said:

> *... we are at an early stage in our understanding of disfluent phenomena. Although studies of disfluencies in a variety of disciplines have provided considerable information about specific aspects of disfluencies, we currently lack the knowledge necessary for an integrated theory.*
>
> [Shriberg, 1994]

to which Junqua & Haton added:

> *In noise-free conditions spontaneous speech understanding for a limited domain, such as database enquiry, is progressively becoming a reality. However, there is still much work to do to handle the phenomena contained in spontaneous speech (e.g. speech repairs and filled pauses).*
>
> [Junqua and Haton, 1996]

It was the goal of the work presented in this thesis to investigate further the use of syntactic knowledge at a level between acoustics and the completed string, within the automatic speech recognition process, with the aim of adding to the knowledge that will allow an integrated theory to be developed in the near future.

The work at Durham deals with repairs at a word lattice level using repair structures identified in an analysis of spontaneous speech. This allows a full grammar to be used to produce the sentence hypotheses from the word lattice and does not rely on the string being produced before analysis takes place. It also does not limit the system by removing information, but simply adds information to the word lattice to allow the hypothesis generation process to choose, using knowledge available to it (e.g. syntax and semantics), the most appropriate path through the lattice. The system does not rely on any pre-identified markers and deals with *cue phrases* and the majority of word fragments automatically.

Chapter 4

Module Analysis

This chapter gives the details of an analysis performed on a sub-section of the automatic speech recognition system, AURAID, being used in the work described in this thesis. The purpose of this analysis was to check the performance of the system By this we mean ensuring that adding knowledge to the system, by incorporating extra modules, increases the performance of the system. From this analysis an *optimum* system could then be formed, on which the work described in the rest of this thesis can be carried out.

4.1 Introduction

The fast, incremental nature of the development of automatic speech recognition systems results in systems comprising of multiple knowledge sources, each performing a specific task. Questions to be addressed include:

- which knowledge sources actually benefit a system?

Using a sub-system of an automatic speech recognition system, currently being developed at the University of Durham, an investigation of each knowledge source used by the system has been undertaken. Each knowledge source was investigated to identify the advantages and disadvantages of using it, and its effect

on the system's overall performance. The original sub-system was modified so that several versions could be easily created. To allow significance testing, each version processed thirty test sentences and measurements were taken of: the search space generated during each run, and the overall performance of the system during each run. Two significance tests (sign test and wilcoxon signed rank test) were used to measure the level of the performance between systems. The system encountered problems when anti-grammar was used and the test sentences contained repairs. It was therefore necessary to ensure that anti-grammar was a significant benefit to the system. The two significance tests were carried out on a comparison of the system with anti-grammar and the system without anti-grammar, processing 100 sentences.

These analyses allowed the optimum system to be identified along with the contribution made by each knowledge source.

4.2 The System

The system, at present, is only part of a final product [Collingham and Garigliano, 1993] [Collingham, 1994]. The front end is a simulated phoneme recogniser with a variable corruption rate of up to 25%. The automatic speech recognition system (AURAID - A University speech Recognition AID) uses the first level of a two level dynamic programming algorithm, incorporating word frequency information, to build a word lattice. A pyramid beam search with a skip & share algorithm is then used to determine sentence hypotheses occurring in the lattice. Anti-grammar rules [Collingham and Garigliano, 1993] are used to select the most appropriate hypotheses.

4.2.1 Phoneme Recognition

The front-end processing of the raw speech signal will be performed by a continuous speech phoneme recognition system. Work of this nature is being performed by a

number of different research groups of which the Defence Research Agency (DRA) and Cambridge University Engineering Department (CUED) are two, but a completed phoneme recogniser was not available at the start of this project (in 1992). To allow development of the post-phonemic processing of interest to this thesis it has been necessary to produce a simulated front-end, in PERL, where, given the actual words spoken a corrupted phoneme string is produced (see figure 4.1). This phoneme string contains realistic corruption to a degree that can be specified by a set of parameters. This could be up to a total of 25%.

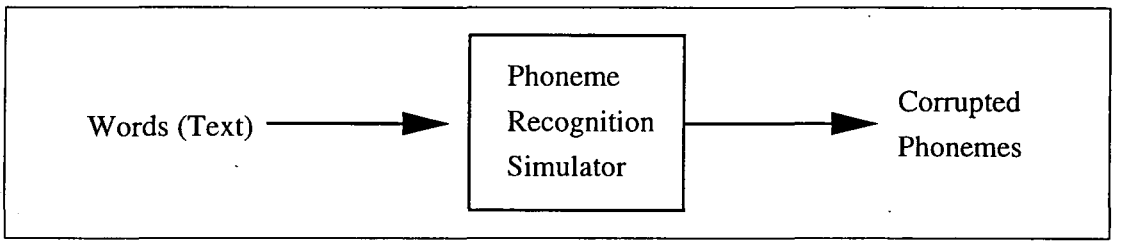


Figure 4.1: The Phoneme Recognition Simulator

The actual corruptions are made with the same distributions as those found in [Browning *et al.*, 1990] which shows the types of insertions, substitutions and deletions that appear in a phoneme recogniser when processing 'real speech'. The research, carried out at the DRA, also gives percentage likelihoods for these modifications (see table 4.1[1]) and the types of modifications that take place (see table 4.2 for the confusion matrix and table 4.3 for the phoneme classes used).

The percentage of insertion, deletion and substitution errors that appear for each phoneme class are used to select, taking the corruption rate supplied into account, the errors that appear within the phoneme string. Then for each phoneme the locations where these errors are going to occur are selected at random. The actual change made (i.e. which phoneme is to replace the selected phoneme) is chosen using random selection while taking the confusion matrix (table 4.2) into account. The random selection process is initiated through a random number seed

[1]Table 4.1 shows that for each phoneme 's' 8.6% of the original phonemes are substituted, 5.6% are deleted and 5.6% of the total number of original phonemes are inserted within the corrupted phoneme string. These figures are for 26.2% phoneme corruption.

which is taken from the time currently held by the system. This random number seed can be used to reproduce an exact copy of the corrupted string using the same input and corruption level.

An example of the corruption procedure for the simulation program (taken from Collingham (1994)) is given below.

Words	for	this	lecture	we're	going
Original Phonemes	f O r	D I s	l E k tS @ r	w I@ r	g @U I N
Corrupted Phonemes	f U@ r	D I s	l E k tS r	w U I@ r	d g @U I N

Words	to	be	looking	at	maintenance	models
Original Phonemes	t @	b i	l U k I N	{ t	m eI n t @ n @ n s	m Q d l z
Corrupted Phonemes	t @	b i	l U g I N	{ t	eI n @ n @ m s	m eI d z

The details of the phoneme corruption are:

```
Number of phonemes: 44

                NUM  SUBS  DELS  INS
plosives    =     8     1     1    1
affrics     =     1     0     0    0
strfrics    =     3     0     0    0
wkfrics     =     2     0     0    0
liquids     =     7     0     1    0
nasals      =     7     1     1    0
vowels      =    16     2     1    1
-----------------------------------------
TOTALS      =           4     4    2 (10)
TOTALS (%)  =         9.1   9.1  4.5 (22.7)
```

After corruption, the example sentence contains 22.7% phoneme error, consisting of 9.1% substitutions, 9.1% deletions and 4.5% insertions.

One main advantage of using a simulated front-end is that it provides reproducible input for the system, which will aid greatly in development and testing. It could be argued that a simulated front-end does not produce a realistic source of knowledge. However, Collingham (1994) argues that the front-end does produce a realistic phoneme string because: the recognition error probabilities were obtained from an existing continuous speech recognition system. The corruption rate is also variable to allow testing of the robustness of the word recognition algorithm with respect to changes in phoneme recognition accuracy.

Phoneme	Sub.	Del.	Ins.	Phoneme	Sub.	Del.	Ins.
s	8.6	5.6	5.6	z	12.9	7.0	4.6
S	1.1	5.4	1.0	Z	12.9	7.0	4.6
f	7.7	4.7	4.3	v	13.3	30.0	14.4
T	11.2	14.0	9.3	D	1.0	1.0	1.0
h	17.9	15.4	5.1	tS	18.5	1.0	1.0
dZ	18.2	9.1	6.1	p	7.3	4.1	27.6
b	17.8	24.2	33.4	t	7.6	6.5	3.3
d	15.9	19.5	20.7	k	3.4	5.5	4.1
g	14.6	4.9	1.0	m	15.6	12.2	8.2
n	15.4	13.8	4.1	N	27.8	20.3	1.9
l	8.4	11.1	7.6	r	2.9	6.2	4.9
w	10.6	3.8	3.0	j	9.5	4.8	4.8
i	11.2	2.7	4.0	I	18.3	9.6	8.1
E	10.8	7.6	2.8	{	10.3	1.2	2.4
A	2.7	1.0	1.8	Q	3.6	7.1	1.0
O	3.6	4.5	0.9	U	1.0	22.2	11.1
u	18.8	6.0	3.0	3	1.0	1.0	22.8
@	16.9	20.9	8.9	V	19.3	8.0	9.1
eI	6.1	1.0	1.0	aI	5.9	2.6	2.3
oI	1.0	1.0	1.0	aU	6.3	4.1	2.1
@U	19.6	3.6	2.4	I@	7.8	1.0	2.0
e@	50.0	1.0	1.0	u@	1.0	1.0	1.0

Table 4.1: The Phoneme Simulator Error Frequency Percents

4.2.2 Word Lattice Generation

A word lattice is a structure, where a set of word hypotheses produced by a phonemic matching process is stored. Recorded with each word hypothesis is its start and end points within the transcription (phoneme string or other spoken utterance representation) and the likelihood of that word having been spoken. So, within the AURAID system, for each phoneme position there are a number of word hypotheses which start at this phoneme. The span of these words is also noted and a score representing the likelihood that that word was actually spoken is kept with the word. This score includes a figure based on the match between the word representation held in the dictionary and the input phoneme string. The score also includes a word frequency value. Word frequency information deals with the number of times a word type is used in normal spoken English and gives a boost to those that are frequent. These details were taken from the Oxford Advanced Learners Dic-

Spoken	Recognised						
	Plo.	Aff.	Strong Fric.	Weak Fric.	Liquid/ Glide	Nas.	Vow.
Plosive	95.9	0.3	0.9	0.8	0.7	0.3	1.1
Affricative	3.5	85.9	7.0	—	—	1.8	1.8
Strong Fricative	1.6	0.4	96.6	0.3	—	0.8	0.3
Weak Fricative	6.1	0.2	0.5	91.7	—	—	1.5
Liquid/Glide	0.8	0.1	—	0.6	94.1	0.4	4.0
Nasal	2.6	—	—	0.4	2.1	90.4	4.5
Vowel	0.7	—	0.1	0.2	0.7	0.5	97.8

Table 4.2: The Phoneme Simulator Confusion Matrix

Class	Name	Phonemes
0	Plosive	p b t d k g
1	Affricative	tS dZ
2	Strong Fricative	s z S Z
3	Weak Fricative	f v T D h
4	Liquid/Glide	l r w j
5	Nasal	n m N
6	Vowel	i I E { A Q O U u 3 V @ aI eI oI aU @U I@ e@ U@

Table 4.3: Phoneme Classes used by AURAID

tionary [Mitton, 1992] which identifies three levels of frequency (common, normal and rare). As well as this information the AURAID system also records the part-of-speech tag that goes with the word. An example of a simple word lattice, that could be produced if the 'Phoneme String', representing the 'Sentence' is corrupted to the level shown in 'Corrupted String', is shown in table 4.5, page 61. Here the word *the* of class ART covers phonemes 1 to 2 and the word *quiz* of class NOUN covers phonemes 13 to 16. A likelihood score is also recorded, but is not shown in this example.

To produce a word lattice a technique for matching word pronunciations against a phoneme string is required. Dynamic programming has become the standard lexical access algorithm for this task. Many systems that use dynamic programming incorporate a multi-pass strategy where initially a cheap (in terms of computational expense) language model is used to identify the likely words, while further passes

bring in more sophisticated, yet expensive, forms of knowledge into the process. It was decided within the AURAID system to separate the dynamic programming algorithm from the contributing knowledge. This allows knowledge to be incorporated into the system without major modifications to the system. It also allows different knowledge sources to work in parallel, on the same search space, rather than allowing one form of knowledge, used in an early dynamic programming pass, to prune the search space and remove hypotheses that would have been strongly favoured by later forms of knowledge. Determining the optimal *serial* combination of different knowledge sources (i.e. which knowledge source should be allowed to prune the search space first) is a complex task. It is more likely that they will need to operate in the same search space and independently of each other so that each knowledge source contributes positively when assessing competing sentence hypotheses. A knowledge source may give bonuses as well as penalties when judging the relative merits of different sentence hypotheses. A sentence hypothesis that is penalised by the grammar may be given a bonus by the semantic knowledge source — a balance needs to be achieved between pruning the large search space and ensuring that the correct hypothesis is not eliminated too early.

AURAID uses dynamic programming to perform word level analysis by matching stored template words, made up of a series of phonemes, with the input phonemes. So that each word in the dictionary is matched against all possible (consecutive) sequences of the input phonemes. The word level analysis algorithm models the kinds of errors which may occur. That is inserted phonemes, deleted phonemes and substituted phonemes. The equations used in the word level analysis algorithm are:

$$
\begin{aligned}
S(w,1,t) = \min\{ \quad & \frac{ins\_pen}{N(w)} + \frac{sub\_pen(w,1,t)}{N(w)} + \min_{r \in R}\{S(r, N(r), t-2)\}; \\
& \frac{sub\_pen(w,1,t)}{N(w)} + \min_{r \in R}\{S(r, N(r), t-1)\}; \\
& \frac{del\_pen}{N(w)} + \frac{sub\_pen(w,1,t)}{N(w)} + \min_{r \in R}\{S(r, N(r)-1, t-1)\}; \\
& \frac{2.0 \times del\_pen}{N(w)} + \frac{sub\_pen(w,1,t)}{N(w)} + \min_{r \in R}\{S(r, N(r)-2, t-2)\} \ \}
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
S(w,2,t) = \min\{ \quad & \frac{ins\_pen}{N(w)} + \frac{sub\_pen(w,2,t)}{N(w)} + S(w,1,t-2); \\
& \frac{sub\_pen(w,2,t)}{N(w)} + S(w,1,t-1); \\
& \frac{del\_pen}{N(w)} + \frac{sub\_pen(w,2,t)}{N(w)} + \min_{r \in R}\{S(r,N(r),t-1)\}; \\
& \frac{2.0 \times del\_pen}{N(w)} + \frac{sub\_pen(w,2,t)}{N(w)} + \min_{r \in R}\{S(r,N(r)-1,t-2)\} \ \}
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
S(w,3,t) = \min\{ \quad & \frac{ins\_pen}{N(w)} + \frac{sub\_pen(w,3,t)}{N(w)} + S(w,2,t-2); \\
& \frac{sub\_pen(w,3,t)}{N(w)} + S(w,2,t-1); \\
& \frac{del\_pen}{N(w)} + \frac{sub\_pen(w,3,t)}{N(w)} + S(w,1,t-1); \\
& \frac{2.0 \times del\_pen}{N(w)} + \frac{sub\_pen(w,3,t)}{N(w)} + \min_{r \in R}\{S(r,N(r),t-2)\} \ \}
\end{aligned}
\tag{4.3}
$$

$$
\begin{aligned}
S(w,p,t) = \min\{ \quad & \frac{ins\_pen}{N(w)} + \frac{sub\_pen(w,p,t)}{N(w)} + S(w,p-1,t-2); \\
& \frac{sub\_pen(w,p,t)}{N(w)} + S(w,p-1,t-1); \\
& \frac{del\_pen}{N(w)} + \frac{sub\_pen(w,p,t)}{N(w)} + S(w,p-2,t-1); \\
& \frac{2.0 \times del\_pen}{N(w)} + \frac{sub\_pen(w,p,t)}{N(w)} + S(w,p-3,t-2) \ \}
\end{aligned}
\tag{4.4}
$$

where $S(w,p,t)$ represents the score for phoneme p of word w when matched against input phoneme t, R is the set of words in the dictionary used by AURAID and $N(r)$ is the length in phonemes of the r'th word. The three penalties, *ins_pen*, *del_pen* and *sub_pen* are based on phoneme class confusions and the error frequencies used within the simulated phoneme corruption process. The phoneme corruption process does have a random factor which selects where the changes are made and what the

changes are.

Equation 4.4 is the general equation used for dynamic programming matching, equations 4.1, 4.2 and 4.3 being for words of phoneme length 1, 2 and 3 respectively. In the general equation, a minimum score choice is taken between: the previous input phoneme being an insertion error; the current input phoneme being correct or a substitution error; or a deletion of the previous phoneme of the current word. In addition, the last line of each equation represents the occurrence of two consecutive deletion errors. For short words, the first three equations perform the same calculation as the general equation but look back at previous words to determine what, if any, error has taken place. Finally, for each input phoneme the end score for each word is adjusted to represent the local score for that word if it were to end at that point in the input.

The word score is calculated by dividing any penalty by the length of the word (in phonemes), so as not to penalise long words too heavily. Each word is, therefore, given a score representing its likelihood of matching a particular sequence of input phonemes. This score is adjusted to take account of the word frequency information. The data structure resulting from the dynamic programming stage is a word lattice.

4.2.3 Dictionary

The disadvantage of using dynamic programming for generating a word lattice is that the time involved is proportional to the size of the dictionary being used: each phoneme of each word in the dictionary is matched against the phoneme input. However, this problem can be reduced by exploiting the fact that this task can be performed in parallel.

There are different dictionaries (see table 4.4) used in the different analyses presented in this thesis. The main dictionary (dictionary 1) currently has 1,985 words of which one hundred and forty six are important words relating to lectures (e.g. "academics", "books", "degree", "journals" and "undergraduates") with the

Dictionary No.	Category Words	LOB Words	Topic Words	Total Words
1	146	1,839		1,985
2	146	354		528
3	146	1,839	290	2,275

Table 4.4: Dictionaries Used in the Analyses within this Thesis

remainder being the 1,839 most frequent words in the LOB corpus. For each word, the system dictionary contains a phoneme pronunciation and one or more syntactic categories (such as VERB, NOUN, etc.). This information was obtained from the Oxford Advanced Learners Dictionary [Mitton, 1992]. A smaller dictionary (dictionary 2) of five hundred and twenty eight words can be used, where three hundred and fifty four were the most frequent words found in the LOB corpus and one hundred and forty six were the important words to lectures. An extended dictionary (dictionary 3) of 2,275 words was also created by including two hundred and ninety topic words, relevant to the lectures analysed in this thesis, to the original 1,985 words of the main dictionary. The actual dictionary used in each analysis will be explained with the analysis details.

For widespread commercial use, the system dictionary clearly needs to be larger, 20,000 words would be a more suitable size, but 1,985 words is adequate for development purposes and also covers the majority of high frequency words used in English. As an illustration, the first two lectures of a second year course on software engineering contained 1,300 unique words, and the first two lectures of a third year course on software engineering contained 1,100 unique words.

Sentence		the	ques-	the	first	question
Phoneme String		D @	k w E s	D @	f 3 s t	k w E s tS @ n
Corrupted String		D @	k w E s	@	f 3 s t	k w E s A tS N
Position		1 2	3 4 5 6	7 8	9 0 1 2	3 4 5 6 7 8 9
Word Lattice	1	the (ART)	question (NOUN)			quest (NOUN) / on (PREP)
	2	there (ADV)			thirst (VERB)	quiz (NOUN) / ton (NOUN)
	3	they (PRON)		the (ART)	fist (NOUN)	question (NOUN)
	4	how (ADV)		turf (NOUN)		square (ADJ) / ton (NOUN)
	5	their (ADJ)	crest (NOUN)		first (ADJ)	

Table 4.5: Example Word Lattice

4.2.4 Word Lattice Parsing

The aim of word lattice parsing is to traverse the word lattice, from left to right or from right to left, and identify all of those paths that span the whole length of the lattice. When a path covers the whole length then it can be seen as a possible interpretation of the original spoken utterance. An example of possible paths through the example word lattice given in table 4.5 are:

the question fist quest on

there crest first question

there question first question

The process of traversing the word lattice is carried out in cycles, where a cycle is when each hypothesis, in the hypothesis list, is expanded into a number of hypotheses by adding one word. The first cycle (presuming a left to right traversal) takes the first words within the lattice and creates a number of hypotheses, known as sentence hypotheses. In table 4.5 there will be five initial hypotheses as five words can start at the first phoneme. In each following cycle, each hypothesis is

expanded one word based on the current end position of the hypothesis and those words, within the word lattice, that start directly after the end position of the hypothesis.

So after the first cycle the hypothesis list would contain the following sentence hypotheses:

```
the
there
they
how
their
```

After the second cycle the hypothesis list would contain the following sentence hypotheses:

```
the question
the crest
there question
there crest
they question
they crest
how
their question
their crest
```

This shows that it was possible to add two words (question and crest) to the end of the sentence hypothesis 'the'. This is also true for the sentence hypotheses 'there', 'they' and 'their'. No words start directly after the sentence hypothesis 'how', within the lattice, therefore, it was not possible to expand this hypothesis. This process continues until the sentence hypotheses reach the end of the word lattice.

The Search

A pure best-first search would expand only the best sentence hypothesis at each stage. This would be an acceptable approach if the phoneme error rate was very low, as it is likely that the (partial) correct sentence would be expanded ahead of all other candidates, and there is no across word boundary knowledge used. When the phoneme error rate is high, however, more sentence hypotheses need to be expanded at each stage during the search through the word lattice to give other sources of knowledge, such as syntax and semantics, a chance of recovering the poorly matching correct sentence.

A beam search could be used to ensure a certain number of hypotheses are followed by expanding those sentence hypotheses that are near the top sentence hypothesis at each stage. Within the AURAID system a pyramid beam search technique is used where in the early stages the number of sentences hypotheses expanded is greater than in latter stages. The beam search makes use of guesses about the score incurred by each sentence hypothesis over the remaining portion of the sentence being processed. These guesses are in the form of underestimates [Winston, 1992]. Each sentence hypothesis in the sentence hypothesis list is a recognition for part of the sentence being processed. The hypotheses all start at the same point but reach to different parts of the sentence being processed. An underestimate score is calculated for each sentence hypothesis based on the remaining amount of the sentence being processed multiplied by some constant determined empirically.

A simple extension of the search thus described would produce a search technique known as A*. The A* search is a best-first (or beam) search that makes use of underestimates of distance remaining as described above, but also discards redundant paths. In other words, if several paths reach the same node in the search, only the best scoring of these paths is kept alive, the others being removed from the search space. The A* is only suitable for word lattice parsing when a no-grammar or unigram language model is used. If there is no grammar (or other knowledge that scores hypotheses across word boundaries) being used within the

system it is not necessary to expand all sentence hypotheses. If two hypotheses end at the same location then only the more likely hypothesis need be expanded as this hypothesis will always result in a more likely completed hypothesis than the less likely sentence hypothesis. This is not the case when across word boundary knowledge is used.

For example, given two sentence hypotheses P, consisting of words p_1, p_2, p_3, and Q, consisting of words q_1, q_2, both hypotheses stretching to node n in the word lattice and with scores 20 and 25 respectively. Using the A* algorithm, we would prune Q because it has a worse score than P — we only keep the best sentence hypothesis that reaches a particular node. If we were to extend P by one more node by adding word w to span the phonemes between node n and node $n+1$, P may now have a score of 30 at node $n+1$. If we had kept Q in the list of sentence hypotheses it may have a lower score even though it would have been extended by the same word, w, because of the grammar penalties (or bonuses) incurred for the new hypotheses P, consisting of words p_1, p_2, p_3, w, and Q, consisting of words q_1, q_2, w.

As the knowledge used in this system can give bonuses or penalties across word boundaries a pyramid beam search technique is used for word lattice parsing. Each sentence hypothesis is given a score so that hypotheses with high scores are unlikely to be correct. The beam search is used to cut out these unlikely hypotheses by only expanding those hypotheses whose score is within a certain percentage of the top scoring hypothesis. A wider beam, higher percentage, is used initially (giving the pyramid effect) to ensure that the correct hypothesis is not eliminated early on in the processing.

The Knowledge

The use of a statistical language model has not been used as the system has been designed to cover unrestricted speech within the lecturing environment. Rather, a set of rules were developed (that were very effective in increasing the performance of the system) that can be used to check the syntactic *incorrectness* of sequences

of words [Collingham and Garigliano, 1993]. These rules are collectively known as an "anti-grammar" because the rules are used to penalise certain syntactic constructs, rather than identify syntactically correct sequences. The rules decrease the likelihood of hypotheses being selected if they contain a certain structural pattern, therefore, removing ill-formed hypotheses from the list of potential hypotheses. Though this is a simplistic form of grammar, in that the version used for the analysis presented here has only fifty eight rules, it has been proven [Collingham and Garigliano, 1993] [Collingham, 1994] that it is very effective in reducing the search space and increasing the performance of a sub-section of an automatic speech recognition system. Our own analysis (see section 4.6.5) shows that anti-grammar is of significant benefit to the system and can therefore be used within the system.

Another technique used in the AURAID system is a skip & share algorithm. This algorithm allows the word lattice parsing system to skip phonemes while traversing the lattice or to allow phonemes to be shared between words. The purpose of this is to overcome phoneme insertions and deletions. Using this technique further sentence hypotheses would be possible, examples (taken from table 4.5) of which would be:

how question fist question

the question thirst quest on

how question first quiz on

The first example shows that the words 'how' and 'question' are allowed to share the 3rd phoneme of the input transcription. The second example shows that the word 'question' and 'thirst' can be joined within a sentence hypotheses even though there is a phoneme between them, within the phoneme transcription. The third example shows a similar procedure with the words 'quiz' and 'on'.

Using these techniques (beam search, skip & share and anti-grammar) an optimum span of the word lattice can be found. This requires using knowledge on phoneme recognition levels, word probabilities and phrase probabilities (as identified by the anti-grammar). The top scoring hypothesis after the final cycle (i.e. the

whole lattice has been traversed) is given as the result of the recognition process.

No.	Phrase
1	It might be something to control the timetable in the university for example.
2	There is within the software industry a concept called the software crisis.
3	Just just so we can document the program.
4	So there are bits in the book that I I I leave out altogether.
5	So it's important to to get this out of your mind.
6	The universities were very happy because they knew it couldn't be written.
7	I unfortunately have undergraduates so so you are my customers my users.
8	I'm trying to manage you in the production of a product.
9	But the the advances have primarily been on the hardware front.
10	It's about managing people, managing projects and managing your software.
11	You should see this book as supplementary reading.
12	I assume that you're reading the relevant sections.
13	That i don't have time to cover.
14	I should also point out that you don't have to know everything that's in this book.
15	It's usually very difficult to maintain.
16	And the sections that i point out each week.
17	And of course since you're all very keen.
18	It basically gives you some background information.
19	The last book is software engineering economics.
20	It's a huge great thick book full of graphs and equations.
21	Instead of delivering the simple tyre on a piece of string.
22	He will tell you real stories about software engineering.
23	This is again a famous set of drawings.
24	It's not just about writing programs.
25	Now just to put this into some sort of perspective.
26	And we can't just write programs and hope they work.
27	There is within the software industry a concept called the software crisis.
28	This has been due to various factors.
29	It's usually very unreliable.
30	Just so we can document the program.

Table 4.6: Sentences Used in the Module Analysis

4.3 The Data

The data used in this analysis was taken from the Durham corpus, which is currently being developed at the University of Durham. From this corpus thirty sentences were selected (see table 4.6) for this analysis. As one of the main interests

of the work presented in this thesis is with speech repairs five of the representative sentences contained speech repairs.

Two dictionaries were also used in the analysis, the 528 and 1,985 word dictionaries. The larger dictionary was used to test the systems performance on a more realistically sized vocabulary. Though 1,985 words is less than the typical vocabulary of a normal person and a system that was to deal with normal speech would require a much larger vocabulary, the 276% increase will give an indication of the effect of increasing the vocabulary size.

4.4 The Analysis

To measure the performance of each knowledge source used within the AURAID system it was necessary to produce different versions of the system, with each version using different combinations of knowledge. These versions were created by building seven switches into the original system.

The purpose of these switches was to allow the different versions to be created easily and to allow the use of the different knowledge sources when required. The switches incorporated into the system were as follows:

1. Narrow beam width (10% & 5%)
2. Medium beam width (20% & 10%)
3. Skip & share algorithm
4. Word frequency information
5. Anti-grammar rules
6. 528 word dictionary
7. 1,985 word dictionary

Not all of these switches are independent therefore they were examined in three phases.

The first two switches (Narrow beam width and Medium beam width) are mutually-exclusive in that they can not be used together. Therefore, the first phase was to identify the optimum beam width. Once the optimum beam width was found it was used by the rest of the systems in this analysis.

Switches 3, 4 and 5 (Skip & Share, Word Frequency and Anti-grammar) are all independent in that they can be switched on and off independently. It was not deemed necessary to examine all combinations of the switches, but it was necessary to see the effect of incorporating a switch into the system. Phase two investigated the inclusion of each of these three switches ensuring that there was a previous version of the system (without the included switch) for comparison purposes.

The final two switches (528 and 1,985 word dictionaries) are also mutually-exclusive. The third phase was to determine the effect of increasing the dictionary size by 276%. Here all of the switches used at the end of phase two were used with the dictionary switches changed to include the larger dictionary.

The switches were used to create seven different systems and the combinations of the switches which made up the different systems can be seen in table 4.7.

System	Switches						
	1	2	3	4	5	6	7
1	x					x	
2		x				x	
3		x	x			x	
4		x		x		x	
5		x	x	x		x	
6		x	x	x	x	x	
7		x	x	x	x		x

Table 4.7: The Switches Used to Produce the Seven Different Systems

The thirty test sentences were processed by each of the seven systems. Data was collected on each cycle. The data collected on each run included: the position of the required hypothesis in the hypothesis list; the hypothesis score of the required

hypothesis; and the score of the top hypothesis.

It is important to note the actual recording mechanism used throughout the majority of the analyses presented in this thesis. Most automatic speech recognition system performances are measured based on the accuracy of the finally selected string and only the words themselves are used in the measurements. However, of most importance to most automatic speech recognition systems is the actual meaning of what has been said and not just the words themselves. This is why the measurements taken in this thesis include the part-of-speech tags. The correct hypothesis is identified by both the words and the tag associated with the word. Consider the example in which the required string was "The red house". If the tags required were "ART ADJ NOUN" and the hypothesis list was as follows:

1. The read house
   ART VERB NOUN
2. The red house
   ADV ADJ NOUN
3. The red house
   ART ADJ NOUN
4. The read house
   ADV VERB NOUN

then the position of the correct hypothesis would be three, even though the string "The red house" appears at position two. This form of measurement can give low results, but it does allow very accurate measurements and comparisons to be made between systems. The effect of including knowledge can be seen directly by the change in position of the required hypothesis within the hypothesis list. A further reason for this more precise form of measurement is that problems associated with repair and speech recognition systems seem to be linked to the use of grammatical knowledge. Therefore, it is important to include grammatical knowledge into the measurements. Also, when semantic knowledge is added to any automatic speech recognition system or the automatic speech recognition system is linked to a natural language processing system then the pre-tagged hypothesis will gain a distinct

advantage over un-tagged sentences. For these reasons it is important to take the grammatical tag of the words into account, as well as the words themselves.

The results of each run were compared to see if the included knowledge had any effect on the performance of the system. Two significance tests (Sign Test and Wilcoxon Signed Rank Test, see section 4.5) were performed on the difference between the systems being compared.

The system encountered problems when anti-grammar was used and the test sentences contained repairs. It was therefore necessary to perform a further analysis to ensure that anti-grammar was a significant benefit to the system. This analysis consisted of performing the two significance tests on an analysis of the comparison of the system with anti-grammar and the system without anti-grammar on 100 sentences.

4.5 Significance Tests

The purpose of these significance tests is to measure the significance of the difference between two systems. No assumptions as to the distribution of results are made and so non-parametric statistical tests are required. Two statistical tests will be performed: the first is the Sign Test which measures the difference between positive and negative outcomes of comparisons between two tests: the second is the Wilcoxon Signed Rank Test which takes into account the magnitude of the difference between the comparisons rather than just the direction (+/-) of the difference.

4.5.1 Sign Test

The Sign Test [Siegel, 1956, Pages 68–75] is used to determine the significance of the sign (+/-) of the differences between pairs of results. It is based on the hypothesis that there are an equal number of positive and negative differences. The difference between two pairs is either positive, negative or zero. Differences of zero can be

either ignored or added to the data as a positive or negative outcome depending on the nature of the test. If the number of the relevant sign is less than or equal to a critical value (which depends on the level of significance you require) then you reject the Null hypothesis that both systems are equal and accept the alternate hypothesis you are trying to measure.

4.5.2 Wilcoxon Signed Rank Test

The Sign Test looks at the direction of the difference between the pairs. The Wilcoxon Signed Rank Test [Siegel, 1956, Pages 75–83] looks at the magnitude of the difference (ignoring the sign) between the pairs. This test, scores the differences based on the rank of the difference between the pairs (i.e. -9 is the same as +9 when the differences are ranked). When a difference between two pairs is the same then the total of the ranks is summed and divided between the pairs (if two differences are 9 and they are ranked 5th and 6th then the rank score for each is 5.5). This ensures that the sum of the parts is always the same for sample populations of the same size. Those ranks that are for negative differences are kept negative and those ranks that are for positive differences are kept position (So two differences of -9 and 9 ranked at positions 5 and 6 would have ranks scores of -5.5 and 5.5 respectively). If the total of the relevant ranks (negative or positive) is less than or equal to a critical value (which depends on the level of significance you require) then you reject the Null hypothesis that both systems are equal and accept the alternate hypothesis you are trying to measure.

4.6 The Results

The results are split into six analyses. The first determines the effect of changing the widths of the pyramid beam search. The second examines the inclusion of the skip & share algorithm. The third examines the inclusion of the word frequency information for sentence selection. The fourth analysis examines the inclusion of the anti-grammar rules and the fifth analysis extends the fourth analysis to 100

Sent	System						
	1	2	3	4	5	6	7
1	44	34	39	16	18	7	7
2	1356	13	13	7	7	1	1
3	–	97	42	13	13	–	–
4	–	–	–	9	9	–	–
5	153	7	7	3	3	–	–
6	–	20	46	3	3	2	13
7	26	5	5	39	47	–	–
8	–	5	5	2	2	2	2
9	29	129	194	1	1	–	–
10	506	47	2142	12	12	1	8
11	15	15	12	6	4	1	1
12	–	67	67	2	2	2	2
13	153	74	81	10	12	5	1
14	–	–	–	470	470	1	1
15	–	112	124	8	8	3	2
16	431	153	213	13	13	3	3
17	68	67	57	2	2	2	2
18	65	3	3	3	3	1	1
19	–	65	129	5	5	1	1
20	–	112	–	106	82	42	32
21	82	13	41	10	10	5	14
22	640	7	7	1	1	1	1
23	–	76	–	21	21	6	3
24	–	57	57	25	25	15	12
25	–	199	199	33	33	7	7
26	65	65	65	38	38	18	12
27	–	37	52	7	7	4	1
28	10	24	24	15	15	11	11
29	135	6	6	6	6	4	4
30	330	2307	–	674	674	4	4

Table 4.8: Final Hypothesis Position of the Required Hypothesis for the Thirty Test Sentences when Processed by the Seven Versions of the System (– indicates the required hypothesis was not present)

repair free sentences. While the final analysis determines the effect of increasing the dictionary from 528 to 1,985 words. Table 4.8 gives the position of the required hypothesis for each of the thirty sentences (Sent) when processed by the seven systems.

4.6.1 Analysis 1 - Beam Widths

The first part of the analysis was concerned with the width(s) used in the beam search. A pyramid beam search is used to ensure that the required hypothesis is not lost early in the processing, before a more detailed analysis can be done. The widths investigated were narrow and medium. The narrow, or highly restricted, beam width expanded those hypotheses whose scores were within 10% of the score of the top hypothesis, for the first three cycles and reduced this to 5% for the remaining cycles. The medium beam width used 20% and 10% as the cut of score percentages.

This first analysis compares the system using a narrow beam width with the system using a medium beam width. Therefore, testing the hypothesis that the system is better when the beam width is increased.

Table 4.9 shows for each sentence (Sent), the position of the required hypothesis when the two systems processed the sentence[2]. It also shows the difference (Diff) between the two positions[3], the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test

The NULL hypothesis will be that both systems are equal.

[2] A position of – shows that the required hypothesis was not in the hypothesis list.

[3] A difference of +* shows that the exact difference can not be calculated but it is positive (i.e. the system with the narrow beam width does not have the required hypothesis in the hypothesis list while the system with the medium beam width does).

The alternate hypothesis will be that the system with a medium beam width is better than the system with a narrow beam width. A one-tail test is therefore appropriate.

Table 4.9 shows that the number of pairs that result in a positive outcome are 23 and the number of pairs that result in a negative outcome are 7. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with a medium beam width is better than the system with a narrow beam width.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 7 negative outcomes, this shows that the system with a medium beam width is significantly better (7 is $\leq$ 7) than the system with a narrow beam width at the p$\ll$0.005 level.

The Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with a medium beam width is better than the system with a narrow beam width. A one-tail test is therefore appropriate.

Table 4.9 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with a medium beam width is better than the system without a narrow beam width. An unknown difference, with the system with a narrow beam width not having the required hypothesis in the hypothesis list (+*), is given a difference of 1 as this is a desirable result. An unknown desirable result is given a minimum difference so as not to favour the unknown difference above its minimum potential. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 384 and the total of the negative ranks is 81.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for p=0.005 is 109. As the analysis has a negative score of 81, this shows that the system with a medium beam width is significantly better (81 is $\leq$ 109) than the system with a narrow beam width at the p$\ll$0.005 level.

4.6.2 Analysis 2 - Skip & Share

This second analysis compares the system using the skip & share (see section 4.2.4, page 65) algorithm with the system not using the skip & share algorithm. Therefore, testing the hypothesis that the system is better when the skip & share algorithm is used.

Table 4.10 shows for each sentence (Sent), the position of the required hypothesis when the two systems processed the sentence. It also shows the difference (Diff) between the two positions[4], the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The first thing to note from the figures in table 4.10 is that the skip & share algorithm does not increase the performance of the system. Therefore, it is not necessary to check to see if the system with the skip & share algorithm is better, but rather check to see if the system with the skip & share algorithm is not better. If this is true then the skip & share algorithm is of little benefit to the system.

The Sign Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with the skip & share algorithm is not better than system the system without the skip & share algorithm. A one-tail

[4]A difference of -* shows that the exact difference can not be calculated but it is negative (i.e. the system using the skip & share algorithm does not have the required hypothesis in the hypothesis list while the system without the skip & share algorithm does).

test is therefore appropriate.

Table 4.10 shows that the number of pairs that result in a positive outcome are 4 and the number of pairs that result in a negative outcome are 26. A difference of zero is a negative outcome as it supports the (negative) alternate hypothesis that the system with the skips & share algorithm is not better than the system without the skip & share algorithm.

Using the sign test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 4 positive outcomes, this shows that the system with the skip & share algorithm is significantly not better than the system without the skip & share algorithm at the $p \ll 0.005$ level.

Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that system the system with the skip & share algorithm is not better than system the system without the skip & share algorithm. A one-tail test is therefore appropriate.

Table 4.10 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it supports the (negative) alternate hypothesis that the system with the skip & share algorithm is not better than the system without the skip & share algorithm. An unknown difference, with the system with the skip & share algorithm not having the required hypothesis in the hypothesis list (-*), is given the maximum rank as this is an undesirable result. For those differences that are negative the results are kept negative while the others are positive.

The total for the positive ranks is 77.5 and the total for the negative ranks is 387.5.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical

value for p=0.005 is 109. As the analysis has a positive score of 77.5, this shows that the system with the skip & share algorithm is significantly not better ($77.5 \leq 109$) than the system without the skip & share algorithm at the $p \ll 0.005$ level.

4.6.3 Analysis 3 - Word Frequency

This third analysis compares the system using word frequency information with the system not using word frequency information. Therefore, testing the hypothesis that the system is better when word frequency information is used.

Table 4.11 shows for each sentence (Sent), the position of the required hypothesis when the two systems processed the sentence. It also shows the difference (Diff) between the two positions, the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with word frequency information is better than the system without word frequency information. A one-tail test is therefore appropriate.

Table 4.11 shows that the number of pairs that result in a positive outcome are 27 and the number of pairs that result in a negative outcome are 3. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with word frequency information is better than the system without word frequency information.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 3 negative outcomes, this shows that the system with word frequency information is significantly better (3 is $\leq$ 7) than the system without

word frequency information at the $p \ll 0.005$ level.

The Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with word frequency information is better than the system without word frequency information. A one-tail test is therefore appropriate.

Table 4.11 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with word frequency information is better than the system without word frequency information. An unknown difference, with the system without word frequency information not having the required hypothesis in the hypothesis list (+*), is given a difference of 1 as this is a desirable result. An unknown desirable result is given a minimum difference so as not to favour the unknown difference above its minimum potential. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 444 and the total of the negative ranks is 21.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for $p=0.005$ is 109. As the analysis has a negative score of 21, this shows that the system with word frequency information is significantly better ($21 \leq 109$) than the system without word frequency information at the $p \ll 0.005$ level.

4.6.4 Analysis 4 - Anti-grammar

This fourth analysis compares the system using anti-grammar with the system not using anti-grammar. Therefore, testing the hypothesis that the system is better when anti-grammar is used.

Table 4.12 shows for each sentence (Sent), the position of the required hypothe-

sis when the two systems processed the sentence. It also shows the difference (Diff) between the two positions, the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is better than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 4.12 shows that the number of pairs that result in a positive outcome are 21 and the number of pairs that result in a negative outcome are 9. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with anti-grammar is better than the system without anti-grammar.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 9 negative outcomes, this shows that the system with anti-grammar is not significantly better (9 is not $\leq$ 7) than the system without anti-grammar at the p$\ll$0.005 level.

The Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is better than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 4.12 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with anti-grammar is better than the system without anti-grammar. An unknown difference, with the system with anti-grammar not

having the required hypothesis in the hypothesis list (-*), is given the maximum rank as this is an undesirable result. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 315 and the total of the negative ranks is 150.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for p=0.005 is 109. As the analysis has a negative score of 150, this shows that the system with anti-grammar is not significantly better (150 is not $\leq$ 109) than the system without anti-grammar at the p$\ll$0.005 level.

Discussion

One problem noted here is the processing of sentences containing repairs. Sentences 3, 4, 5, 7 and 9 all contain repairs and this seems to cause problems when the system using anti-grammar is processing the sentences.

If these were removed from this analysis:

For the Sign Test with a population of 25 the critical value for p=0.005 is 5. The analysis would have 4 negative comparisons, showing that the system with anti-grammar would be significantly better than the system without anti-grammar, when processing sentences which don't contain repairs, at the p$\ll$0.005 level.

For the Wilcoxon Signed Rank Test with a population of 25 the critical value for p=0.005 is 68. The analysis would have a total negative score of 10, showing that the system with anti-grammar would be significantly better than the system without anti-grammar, when processing sentences which don't contain repairs, at the p$\ll$0.005 level.

As the speech we are working on contains repairs we need to be sure that anti-grammar processing is worth using. A further analysis comparing these two systems in 100 examples, without repairs, has been carried out to ensure that anti-grammar is of benefit to the system when processing speech not containing repairs.

4.6.5 Analysis 5 - Anti-grammar Extension

This fifth analysis compares the system using anti-grammar with the system not using anti-grammar on a further 100 repair free sentences.

Table 4.13 shows for each sentence (Sent), the position of the required hypothesis when the system with anti-grammar (AG) and the system without anti-grammar (No-AG) processed the 100 sentences. It also shows the difference (Diff) between the two positions and the direction of the difference (+/-). Table 4.14 shows for each sentence (Sent) the rank (Rank) of the difference (Diff), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is better than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 4.13 shows that the number of pairs that result in a positive outcome are 87 and the number of pairs that result in a negative outcome are 13. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with anti-grammar is better than the system without anti-grammar.

Using the Sign Test with a sample population of 100 the critical value for p=0.005 is 32. As the analysis has 13 negative outcomes, this shows that the system with anti-grammar is significantly better (13 is $\leq$ 32) than the system without anti-grammar at the p$\ll$0.005 level.

The Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is better

than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 4.14 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with anti-grammar is better than the system without anti-grammar. For this analysis those pairs that can not result in any difference (i.e. the system without anti-grammar does not have the required hypothesis in the hypothesis list, 11 cases) are removed. This only removes examples that are going to benefit the system therefore, we are not increasing the likelihood of a desirable result. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 3,741 and the total of the negative ranks is 264.

Using the Wilcoxon Signed Rank Test with a sample population of 89[5] the critical value for p=0.005 is 1,410. As the analysis has a negative score of 264, this shows that the system with anti-grammar is significantly better (264 is $\leq$ 1,410) than the system without anti-grammar at the p$\ll$0.005 level.

4.6.6 Analysis 6 - Increased Dictionary

This sixth analysis compares the final system processing the thirty test sentences with the small dictionary (528 words) and the larger dictionary (1,985 words). Therefore, testing the hypothesis that the system is not worse when the larger dictionary (an increase of 276%) is used.

Table 4.15 shows for each sentence (Sent), the position of the required hypothesis when the system processed the sentences using the two dictionaries. It also shows the difference (Diff) between the two positions, the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed

[5]The original 100 less the 11 removed.

Rank) used in the Wilcoxon Signed Rank test.

The Sign Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system using the larger dictionary will not be worse than the system using the smaller dictionary. A one-tail test is therefore appropriate.

Table 4.15 shows that the number of pairs that result in a positive outcome are 27 and the number of pairs that result in a negative outcome are 3. A difference of zero is a positive outcome as it supports the (negative) alternate hypothesis that the system using the larger dictionary is not worse than the system using the smaller dictionary.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 3 negative outcomes, this shows that the system using the larger dictionary is significantly not worse (3 is $\leq$ 7) than the system using the smaller dictionary at the p$\ll$0.005 level.

The Wilcoxon Signed Rank Test

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system using the larger dictionary will not be worse than the system using the smaller dictionary. A one-tail test is therefore appropriate.

Table 4.15 shows the position (Rank) for each sentence (Sent) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a positive outcome as it supports the (negative) alternate hypothesis that the system using the larger dictionary is not worse than the system using the smaller dictionary. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 380 and the total of the negative ranks is 85.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for p=0.005 is 109. As the analysis has a negative score of 85, this shows that the system using the larger dictionary is significantly not worse (85 is $\leq$ 109) than the system using the smaller dictionary at the p$\ll$0.005 level.

4.6.7 Results Summary

Generally, the system showed an increased performance, using the position of the required hypothesis in the hypothesis list, when knowledge sources were added.

These analysis identified two main problems with the current system. The first is that the skip & share algorithm did not improve the performance of the system. The analyses showed significantly that the system with the skip & share algorithm is not better than the system not using the skip & share algorithm.

The second is that speech repairs cause a major problem when the anti-grammar rules are used. The anti-grammar has been produced to punish those hypotheses that have ill-formed structures. It is well known that repairs are a definite problem found in spontaneous speech and it is expected [Hindle, 1983] that repairs do not follow normal constructs of natural speech. As repairs are ill-formed structures and anti-grammar punishes ill-formed structures, it is not unexpected that a system using anti-grammar (not taking account of repairs) would result in a decreased performance when processing sentences containing repairs. The anti-grammar rules themselves are of significant benefit to the system when it is processing repair free speech so it is not desirable to remove the anti-grammar rules from the system. It is therefore necessary to have some process that deals specifically with repairs before/while the anti-grammar is being used.

Sent	Narrow Beam Width	Medium Beam Width	Diff	+/-	Rank	Signed Rank
4	–	–	-0	-	1	-2.5
11	15	15	-0	-	2	-2.5
26	65	65	-0	-	3	-2.5
14	–	–	-0	-	4	-2.5
3	–	97	+*	+	5	10.5
6	–	20	+*	+	6	10.5
8	–	5	+*	+	7	10.5
12	–	67	+*	+	8	10.5
15	–	112	+*	+	9	10.5
17	68	67	1	+	10	10.5
19	–	65	+*	+	11	10.5
20	–	112	+*	+	12	10.5
23	–	76	+*	+	13	10.5
24	–	57	+*	+	14	10.5
25	–	199	+*	+	15	10.5
27	–	37	+*	+	16	10.5
1	44	34	10	+	17	17
28	10	24	-14	-	18	-18
7	26	5	21	+	19	19
18	65	3	62	+	20	20
21	82	13	69	+	21	21
13	153	74	79	+	22	22
9	29	129	-100	-	23	-23
29	135	6	129	+	24	24
5	153	7	146	+	25	25
16	431	153	278	+	26	26
10	506	47	459	+	27	27
22	640	7	633	+	28	28
2	1356	13	1343	+	29	29
30	330	2307	-1970	-	30	-30

Table 4.9: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system with a Narrow Beam Width and the system with a Medium Beam Width

Sent	Without Skip & Share	With Skip & Share	Diff	+/-	Rank	Signed Rank
2	13	13	-0	-	1	-7.5
4	–	–	-0	-	2	-7.5
5	7	7	-0	-	3	-7.5
7	5	5	-0	-	4	-7.5
8	5	5	-0	-	5	-7.5
12	67	67	-0	-	6	-7.5
14	–	–	-0	-	7	-7.5
18	3	3	-0	-	8	-7.5
22	7	7	-0	-	9	-7.5
24	57	57	-0	-	10	-7.5
25	199	199	-0	-	11	-7.5
26	65	65	-0	-	12	-7.5
28	24	24	-0	-	13	-7.5
29	6	6	-0	-	14	-7.5
11	15	12	3	+	15	15
1	34	39	-5	-	16	-16
13	74	81	-7	-	17	-17
15	112	124	-10	-	18	-18.5
17	67	57	10	+	19	18.5
27	37	52	-15	-	20	-20
6	20	46	26	+	21	21
21	13	41	-28	-	22	-22
3	97	42	55	+	23	23
16	153	213	-60	-	24	-24
19	65	129	-64	-	25	-25
9	129	194	-65	-	26	-26
10	47	2142	-2095	-	27	-27
23	76	–	-*	-	28	-28
20	112	–	-*	-	29	-29
30	2307	–	-*	-	30	-30

Table 4.10: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system without Skip & Share processing and the system with Skip & Share processing

Sent	Without Word Frequency	With Word Frequency	Diff	+/-	Rank	Signed Rank
18	3	3	-0	-	1	-1.5
29	6	6	-0	-	2	-1.5
4	–	9	+*	+	3	3.5
14	–	470	+*	+	4	3.5
8	5	2	3	+	5	5.5
21	13	10	3	+	6	5.5
5	7	3	4	+	7	7
2	13	7	6	+	8	9
20	112	106	6	+	9	9
22	7	1	6	+	10	9
11	15	6	9	+	11	11.5
28	24	15	9	+	12	11.5
6	20	3	17	+	13	13
1	34	16	18	+	14	14
24	57	25	27	+	15	15.5
26	65	38	27	+	16	15.5
27	37	7	30	+	17	17
7	5	39	-34	-	18	-18
10	47	12	35	+	19	19
23	76	21	55	+	20	20
19	65	5	60	+	21	21
13	74	10	64	+	22	22
12	67	2	65	+	23	23.5
17	67	2	65	+	24	23.5
3	97	13	84	+	25	25
15	112	8	104	+	26	26
9	129	1	128	+	27	27
16	153	13	140	+	28	28
25	199	33	166	+	29	29
30	2307	674	1633	+	30	30

Table 4.11: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system without Word Frequency processing and the system with Word Frequency processing

Sent	Without Anti-grammar	With Anti-grammar	Diff	+/-	Rank	Signed Rank
8	2	2	-0	-	1	-2.5
12	2	2	-0	-	2	-2.5
17	2	2	-0	-	3	-2.5
22	1	1	-0	-	4	-2.5
6	3	2	1	+	5	5
18	3	1	2	+	6	6.5
29	6	4	2	+	7	6.5
11	4	1	3	+	8	8.5
27	7	4	3	+	9	8.5
19	5	1	4	+	10	10.5
28	15	11	4	+	11	10.5
15	8	3	5	+	12	12.5
21	10	5	5	+	13	12.5
2	7	1	6	+	14	14
13	12	5	7	+	15	15
16	13	3	10	+	16	16.5
24	25	15	10	+	17	16.5
1	18	7	11	+	18	18.5
10	12	1	11	+	19	18.5
23	21	6	15	+	20	20
26	38	18	20	+	21	21
25	33	7	26	+	22	22
20	82	42	40	+	23	23
14	470	1	469	+	24	24
30	674	4	670	+	25	25
3	13	–	-*	-	26	-28
4	9	–	-*	-	27	-28
5	3	–	-*	-	28	-28
7	47	–	-*	-	29	-28
9	1	–	-*	-	30	-28

Table 4.12: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system without Anti-grammar processing and the system with Anti-grammar processing

Sent	No-AG	AG	Diff	+/-	Sent	No-AG	AG	Diff	+/-
1	2	2	-0	-	2	3	1	2	+
3	36	1	35	+	4	5	5	-0	-
5	10	4	6	+	6	10	5	5	+
7	3581	17	3564	+	8	—	4	+*	+
9	13	3	10	+	10	81	96	-15	-
11	3	1	2	+	12	6	39	-33	-
13	3	1	2	+	14	—	261	+*	+
15	9	2	7	+	16	22	6	16	+
17	3	2	1	+	18	820	220	600	+
19	14	2	12	+	20	90	33	57	+
21	3	1	2	+	22	168	30	138	+
23	15	3	12	+	24	1	1	-0	-
25	—	—	-0	-	26	27	8	19	+
27	31	135	-104	-	28	808	16	792	+
29	33	13	20	+	30	7	1	6	+
31	11	9	2	+	32	2	1	1	+
33	56	25	31	+	34	25	15	10	+
35	3	3	-0	-	36	—	18	+*	+
37	31	4	27	+	38	15	11	4	+
39	3	2	1	+	40	8	3	5	+
41	11	9	2	+	42	57	26	31	+
43	7	2	5	+	44	—	1	+*	+
45	5	5	-0	-	46	1	1	-0	-
47	28	8	20	+	48	1416	14	1402	+
49	3	2	1	+	50	5	2	3	+
51	32	10	22	+	52	—	22	+*	+
53	27	5	22	+	54	1	1	-0	-
55	6	2	4	+	56	2	1	1	+
57	15	9	6	+	58	45	5	40	+
59	2	1	1	+	60	8	1	7	+
61	5	2	3	+	62	—	41	+*	+
63	1491	1	1490	+	64	26	1	25	+
65	41	7	34	+	66	1288	1	1287	+
67	49	23	26	+	68	4	1	3	+
69	8	3	5	+	70	12	3	9	+
71	6	3	3	+	72	17	5	12	+
73	2	1	1	+	74	19	1	18	+
75	7	6	1	+	76	106	21	85	+
77	3	1	2	+	78	—	19	+*	+
79	—	5	+*	+	80	32	20	12	+
81	22	6	16	+	82	1	1	-0	-
83	11	9	2	+	84	21	17	4	+
85	25	3	22	+	86	3	1	2	+
87	—	4	+*	+	88	2	1	1	+
89	19	9	10	+	90	3	1	2	+
91	1	1	-0	-	92	—	14	+*	+
93	29	6	23	+	94	—	70	+*	+
95	196	84	112	+	96	46	22	24	+
97	17	1	16	+	98	35	18	17	+
99	67	33	34	+	100	27	3	24	+

Table 4.13: The Results of the Extended Comparison of the System Using Anti-grammar (AG) and the System Not Using Anti-grammar (No-AG).

Rank	Diff	Signed Rank	Sent	Rank	Diff	Signed Rank	Sent
1	-0	-5.5	1	51	12	51.5	23
2	-0	-5.5	4	52	12	51.5	72
3	-0	-5.5	24	53	12	51.5	80
4	-0	-5.5	25	54	-15	-54	10
5	-0	-5.5	35	55	16	56	16
6	-0	-5.5	45	56	16	56	81
7	-0	-5.5	46	57	16	56	97
8	-0	-5.5	54	58	17	58	98
9	-0	-5.5	82	59	18	59	74
10	-0	-5.5	91	60	19	69	26
11	1	15	17	61	20	61.5	29
12	1	15	32	62	20	61.5	47
13	1	15	39	63	22	64	51
14	1	15	49	64	22	64	53
15	1	15	56	65	22	64	85
16	1	15	59	66	23	66	93
17	1	15	73	67	24	67.5	96
18	1	15	75	68	24	67.5	100
19	1	15	88	69	25	69	64
20	2	24.5	2	70	26	70	67
21	2	24.5	11	71	27	71	37
22	2	24.5	13	72	31	72.5	33
23	2	24.5	21	73	31	72.5	42
24	2	24.5	31	74	-33	-74	12
25	2	24.5	41	75	34	75.5	65
26	2	24.5	77	76	34	75.5	99
27	2	24.5	83	77	35	77	3
28	2	24.5	86	78	40	78	58
29	2	24.5	90	79	57	79	20
30	3	31.5	50	80	85	80	76
31	3	31.5	61	81	-104	-81	27
32	3	31.5	68	82	112	82	95
33	3	31.5	71	83	138	83	22
34	4	35	38	84	600	84	18
35	4	35	55	85	792	85	28
36	4	35	84	86	1287	86	66
37	5	38.5	6	87	1402	87	48
38	5	38.5	40	88	1490	88	63
39	5	38.5	43	89	3564	89	7
40	5	38.5	69	90	+*	–	8
41	6	42	5	91	+*	–	14
42	6	42	30	92	+*	–	36
43	6	42	57	93	+*	–	44
44	7	44.5	15	94	+*	–	52
45	7	44.5	60	95	+*	–	62
46	9	46	70	96	+*	–	78
47	10	48	9	97	+*	–	79
48	10	48	34	98	+*	–	87
49	10	48	89	99	+*	–	92
50	12	51.5	19	100	+*	–	94

Table 4.14: The Ranks and Scores used in the Extended Comparison of the System Using Anti-grammar and the System Not Using Anti-grammar.

Sent	Small Dictionary	Larger Dictionary	Diff	+/-	Rank	Signed Rank
1	7	7	0	+	1	10.5
2	1	1	0	+	2	10.5
3	–	–	0	+	3	10.5
4	–	–	0	+	4	10.5
5	–	–	0	+	5	10.5
7	–	–	0	+	6	10.5
8	2	2	0	+	7	10.5
9	–	–	0	+	8	10.5
11	1	1	0	+	9	10.5
12	2	2	0	+	10	10.5
14	1	1	0	+	11	10.5
16	3	3	0	+	12	10.5
17	2	2	0	+	13	10.5
18	1	1	0	+	14	10.5
19	1	1	0	+	15	10.5
22	1	1	0	+	16	10.5
25	7	7	0	+	17	10.5
28	11	11	0	+	18	10.5
29	4	4	0	+	19	10.5
30	4	4	0	+	20	10.5
15	3	2	1	+	21	21
23	6	3	3	+	22	23
24	15	12	3	+	23	23
27	4	1	3	+	24	23
13	5	1	4	+	25	25
26	18	12	6	+	26	26
10	1	8	-7	-	27	-27
21	5	14	-9	-	28	-28
20	42	32	10	+	29	29
6	2	13	-11	-	30	-30

Table 4.15: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system using the small Dictionary and the system using the large Dictionary

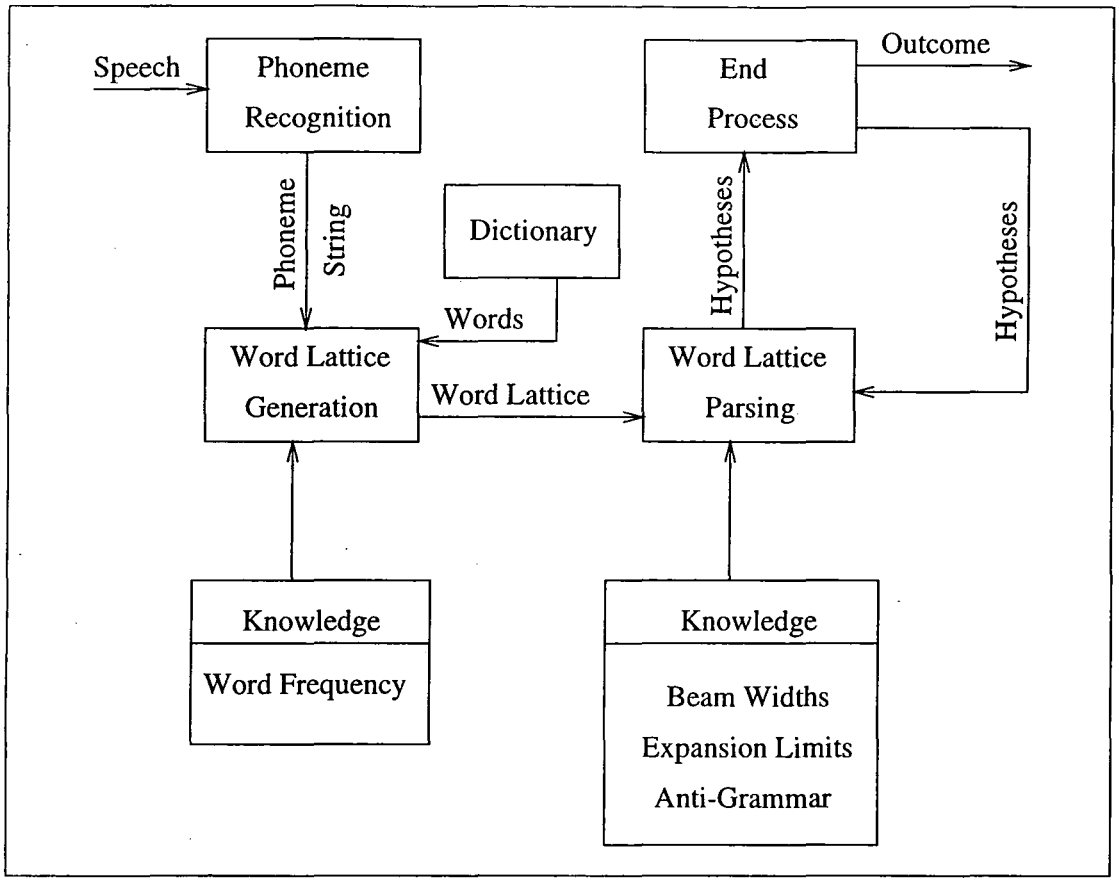


Figure 4.2: Block Structure of AURAID

4.7 The Base System

From this analysis a base system has been produced that will be used throughout this thesis. The system [Collingham, 1994], at present, is only a sub-section of a final product but its performance shows that further expansions will lead to a practical automatic speech recognition system. The remainder of this section outlines the base system for use in the rest of this thesis and a diagram showing how the sections interact can be seen in figure 4.2.

The base system makes use of the pyramid beam search and the analyses given above indicates that the best performance is given when the widths of the beam search are set to 20% initially and then to 10% after three cycles. The system also makes use of word frequency information taken from the Oxford Advanced Learners Dictionary [Mitton, 1992] and also the anti-grammar rules [Collingham

and Garigliano, 1993] produced during the development of the AURAID system. The original system made use of a skip & share algorithm, but the analyses above showed that this did not improve the performance of the system, therefore, the skip & share algorithm has been removed from the base system.

The analyses also showed, significantly, that the system using the larger dictionary was not worse than the system using the smaller dictionary. Therefore, the larger dictionary could be used in the rest of the analyses in this thesis.

4.8 Chapter Summary

This chapter presents an analysis of a sub-section of an automatic speech recognition system which is being developed at the University of Durham. Seven versions of the system, created by including different sources of knowledge, processed thirty test sentences from a lecture presented at the University of Durham and recorded as part of the Durham corpus.

The measurements taken for this analysis were very accurate, using the part-of-speech tags (VERB, NOUN, etc.) of the words to identify the location of the required hypothesis within a hypothesis list. The comparisons between the seven different system formats showed that the following were of benefit: a medium beam width (20% and 10%), word frequency information and anti-grammar rules. The skip & share algorithm did not show any benefit to the system's performance. The analysis also showed that using an increased vocabulary was not detrimental to the system.

This work gives identifies those parts of the system that are of benefit to the system and thus a base system that can be built upon. The production of further modules dealing with such things as speech repairs and semantics is now possible.

Chapter 5

Repair Analysis

This chapter presents an analysis of the effect of repairs, contained in spontaneous speech, on the performance of the base system described in chapter 4. In a previous analysis (see chapter 4) it was noted that speech repairs caused problems for the system and decreased the system's performance. The aim of this analysis is to identify the extent of the problems posed by repairs.

5.1 Introduction

Repairs are a part of spontaneous speech which cause problems for automatic speech recognition systems. It is expected that repairs do not exactly follow the normal constructs of natural speech since spontaneous speech is:

> *... a mixed set of apparently grammatical and ungrammatical strings.*
>
> [Hindle, 1983]

An analysis on the effect of different knowledge sources on the performance of the automatic speech recognition system (see chapter 4) demonstrated that a problem was encountered when grammatical knowledge was introduced into a system attempting to analyse speech containing a repair. This suggests that speech repairs do not use the structures of natural speech.

The analysis shows that a grammar alone can not deal with all the problems of spontaneous speech and further processing in combination with the grammar is required to overcome the problems of speech repairs and produce acceptable recognition results.

5.2 The Data

From the one hundred and seven speech repairs found in the Durham data (see chapter 7) thirty were selected for this analysis (see table 5.1). These repairs were selected to cover all three repair types (types 1, 2 and 3, see section 7.3, page 131) and to include a range of repair sizes. Each of the selected passages were reduced in size, to concentrate on the repair. As word fragments and *cue phrases* are a part of spontaneous speech repairs, four of the thirty repairs contained word fragments (passages 2, 9, 10 and 28) and one of the selected repairs contained a *cue phrase* (passage 7). The word fragments were contained in two type 1, one type 2 and one type 3 repair while the *cue phrase* was contained in a type 1 repair.

Thirty control passages (see table 5.2) were also processed, so that a comparison between repair passages and non-repair passages could be undertaken. These thirty passages were the thirty disfluent passages with the repairs corrected.

This gave sixty test passages with which to test the AURAID system on its performance with respect to speech repairs.

The vocabulary used in this analysis was the 1,985 word dictionary (see section 4.2.3 and table 4.4).

5.3 The Analysis

Two versions of the base AURAID system were created for this analysis. Both systems used the pyramid beam search to produce sentence hypotheses and word frequency information to select the most appropriate hypothesis. But only one

No.	Repair Phrase	Repair Type
1	in fact **the** | **the** book	1
2	hold **complete inf-** | **complete information** about a project	1
3	hard **to** | **to** convert from that	1
4	now **you can** | **you can** see	1
5	not **about** | **just for** writing programs	3
6	the same **programs** | **sets of programs**	2
7	describing **the** | **if you like the** central part	1
8	like **a** | **a** typical economics book	1
9	**the ques-** | **the first question** to answer is	2
10	**they usuall-** | **it usually** costs more	3
11	it is **the** | **the** thing called	1
12	i can **actually** | **finally** get round to writing	3
13	a lot of time **and** | **and** we were spending	1
14	he will tell you **stories** | **real stories** that show	2
15	a thousand **people working** | **programmers working** on the project	3
16	it is important **to** | **to** get this out of your mind	1
17	**this book** | **this course** is not about research	3
18	**just so** | **just so** we can document the program	1
19	there are bits in the book that **i** | **i** | **i** leave out altogether	1
20	between small **or** | **and** large software projects	3
21	i unfortunately have undergraduates **so** | **so** you are my customers	1
22	**the** | **the** advances have primarily been on the hardware front	1
23	the programs you've been **written** | **writing** allow you to build kites	3
24	the third one **on** | **on** the list	1
25	full of graphs and equations **and** | **and** unusually makes sense	1
26	i've written three **up** | **up** there	1
27	lose track of **the** | **the size** | **the sheer size** of our code	2
28	a very **import-** | **important** aspect **of** | **of** software engineering	1
29	**hardware** | **computer hardware** has got better	2
30	**in my eyes** | **in my mind** this **is** | **is** what i think	3

Table 5.1: Phrases Used in the Repair Analysis

No.	Control Phrase
1	in fact **the** book
2	hold **complete information** about a project
3	hard **to** convert from that
4	now **you can** see
5	not **just for** writing programs
6	the same **sets of programs**
7	describing **the** central part
8	like **a** typical economics book
9	**the first question** to answer is
10	**it usually** costs more
11	it is **the** thing called
12	i can **finally** get round to writing
13	a lot of time **and** we were spending
14	he will tell you **real stories** that show
15	a thousand **programmers working** on the project
16	it is important **to** get this out of your mind
17	**this course** is not about research
18	**just so** we can document the program
19	there are bits in the book that **i** leave out altogether
20	between small **and** large software projects
21	i unfortunately have undergraduates **so** you are my customers
22	**the** advances have primarily been on the hardware front
23	the programs you've been **writing** allow you to build kites
24	the third one **on** the list
25	full of graphs and equations **and** unusually makes sense
26	i've written three **up** there
27	lose track of **the sheer size** of our code
28	a very **important** aspect **of** software engineering
29	**computer hardware** has got better
30	**in my mind** this **is** what i think

Table 5.2: Control Phrases Used in the Repair Analysis

used the anti-grammar rules, in conjunction with the word-frequency information, to help select the most appropriate hypothesis.

Corruption within the phoneme generation system was set to 15%.

Both systems processed the thirty disfluent passages with the current hypothesis list recorded after every cycle. The position of the required hypothesis within the list of hypotheses, the score of the required hypothesis and the score of the top hypothesis in the list was recorded.

It is important, once again, to note the accuracy of the measurements used within this analysis. As with the previous analysis the measurements include both the part-of-speech tag and the word itself. Therefore, the exact effect of the inclusion of the anti-grammar rules can be seen on the required output. See section 1.3 and section 4.4 for more details of this measurement criteria.

A comparison was made between the results of the two systems on the disfluent passages to see if the inclusion of the anti-grammar rules gave a performance increase or decrease. The comparison was based on the position of the required hypothesis in the hypothesis list. The further up the list the better the position of the hypothesis. This measure was used as the anti-grammar does not decrease the score of good hypotheses but increases the score of bad hypotheses. It may not be that the required hypothesis is closer to the top through its score, but is more likely to be closer to the top in position.

The thirty repaired passages were run through the second version of the system, where the system used the anti-grammar rules, and the same information as with the disfluent passages was recorded. This information was compared with the information from the equivalent disfluent passages (i.e. the passage before the repair was corrected) so that it could be checked to see whether the disfluent passages were causing the problems rather than the anti-grammar having problems with the normal constructs of the passages.

repair type	increase	no change	decrease	totals
1		3	14	17
2		1	4	5
3	2	1	5	8
totals	2	5	23	30

Table 5.3: Repair Type and System Performance: comparing the system without anti-grammar with the system with anti-grammar when processing passages containing repairs

5.4 The Results

5.4.1 System Comparison

Table 5.3 shows a breakdown of repair types and the performance of the system with respect to these repair types. It shows that of the seventeen passages containing type 1 repairs, fourteen showed a decrease in performance when anti-grammar was used, and three showed neither an increase or decrease in performance.

Of the thirty disfluent passages analysed twenty three showed a decrease in performance when the anti-grammar rules were used, five showed no difference and only two showed any improvement.

Table 5.4 shows for each passage (Pass), the position of the required hypothesis when the two systems processed the passage (a position of — shows that the required hypothesis was not in the hypothesis list). It also shows the difference (Diff) between the two positions (a difference of -* shows that the exact difference can not be calculated but it is negative (i.e. the system with anti-grammar does not have the required hypothesis in the hypothesis list while the system without anti-grammar does)), the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

Pass	Without Anti-grammar	With Anti-grammar	Diff	+/-	Rank	Signed Rank
2	—	—	0	+	1	3
4	5	5	0	+	2	3
9	—	—	0	+	3	3
10	—	—	0	+	4	3
28	—	—	0	+	5	3
15	13	4	9	+	6	6
17	37	12	24	+	7	7
26	24	74	-50	-	8	-8
29	1	54	-53	-	9	-9
6	106	193	-87	-	10	-10
5	13	156	-143	-	11	-11
12	8	153	-145	-	12	-12
19	5	166	-161	-	13	-13
24	21	596	-575	-	14	-14
30	36	1254	-1218	-	15	-15
1	1	—	-*	-	16	-23
3	4	—	-*	-	17	-23
7	2	—	-*	-	18	-23
8	3	—	-*	-	19	-23
11	1	—	-*	-	20	-23
13	3	—	-*	-	21	-23
14	16	—	-*	-	22	-23
16	5	—	-*	-	23	-23
18	27	—	-*	-	24	-23
20	2	—	-*	-	25	-23
21	9	—	-*	-	26	-23
22	9	—	-*	-	27	-23
23	3	—	-*	-	28	-23
25	1	—	-*	-	29	-23
27	7	—	-*	-	30	-23

Table 5.4: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system without anti-grammar and the system with anti-grammar

The Sign Test[1]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is worse than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 5.4 shows that the number of pairs that result in a positive outcome are 7 and the number of pairs that result in a negative outcome are 23. A difference of zero is a positive outcome as it does not support the alternate hypothesis that the system with anti-grammar is worse than the system without anti-grammar.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 7 positive outcomes, this shows that the system with anti-grammar is significantly worse (7 is $\leq$ 7) than the system without anti-grammar at the p$\ll$0.005 level, when processing passages containing repairs.

The Wilcoxon Signed Rank Test[2]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with anti-grammar is worse than the system without anti-grammar. A one-tail test is therefore appropriate.

Table 5.4 shows the position (Rank) for each passage (Pass) when the difference (Diff) between the pairs is used to rank the passages. As with the Sign Test a difference of zero is a positive outcome as it does not support the alternate hypothesis that the system with anti-grammar is worse then the system without anti-grammar. An unknown difference, with the system with anti-grammar not having the required hypothesis in the hypothesis list (-*), is given the maximum rank as this is an undesirable result. For those differences that are negative the results are kept negative while the others are positive.

[1]See section 4.5, page 70, for a description of the Sign Test.
[2]See section 4.5, page 70, for a description of the Wilcoxon Signed Rank Test.

The total of the positive ranks is 28 and the total of the negative ranks is 437.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for p=0.005 is 109. As the analysis has a positive score of 28, this shows that the system with anti-grammar is significantly worse (28 is $\leq$ 109) than the system without anti-grammar at the p$\ll$0.005 level, when processing passages containing repairs.

These tests demonstrates that anti-grammar rules are of little benefit to an automatic speech recognition system when the passages it is processing contain repairs. From this we can deduce that speech repairs have the potential to cause problems to automatic speech recognition systems using anti-grammar.

Passage Analysis

As an illustration of the types of problems that can occur the results of the comparisons for three of the repair passages are given in table 5.6. Table 5.5 gives an index of the headings for this table and the rest of the tables in this chapter. Passage 1 (phrase 8) has a repeated article ("a") and shows that this alone can cause problems when anti-grammar rules are used. Passage 2 (phrase 7) contains a *cue phrase* ("if you like") and Passage 3 (phrase 9) contains a word fragment ("ques-") and an inserted word ("first").

Heading	Description
word	The word the required hypothesis is up to.
top s	The score for the top hypothesis in the hypothesis list.
p	The position of the required hypothesis in the hypothesis list.
score	The score of the required hypothesis in the hypothesis list.
% dif	Percentage difference between the two scores ("top s" & "score").
s dif	The difference between the "% dif" of the system using no anti-grammar and the "% dif" of the system using anti-grammar.
p dif	The difference between the position of the system using no anti-grammar and the position of the system using anti-grammar.

Table 5.5: Index for Tables 5.6, 5.9 and 5.10

Passage 1 : like a | a typical economics book										
	No Anti Grammar				*With Anti Grammar*				totals	
word	*top s*	*p*	*score*	*% dif*	*top s*	*p*	*score*	*% dif*	*s dif*	*p dif*
like	76.0	2	76.0	0	76.0	2	76.0	0	0	0
a	76.1	17	80.0	5	76.1	10	80.0	5	0	7
a	73.1	8	76.0	4	69.5	5404	176.0	153	-149	-5396
typical	54.1	20	68.6	27					-	-
Passage 2 : describing the | if you like the central part										
	No Anti Grammar				*With Anti Grammar*				totals	
word	*top s*	*p*	*score*	*% dif*	*top s*	*p*	*score*	*% dif*	*s dif*	*p dif*
describing	92.4	1	92.4	0	92.4	1	92.4	0	0	0
the	84.4	1	84.4	0	84.4	1	84.4	0	0	0
if	72.4	15	92.5	26	72.4	384	191.5	165	-138	-369
Passage 3 : the ques- | the first question to answer										
	No Anti Grammar				*With Anti Grammar*				totals	
word	*top s*	*p*	*score*	*% dif*	*top s*	*p*	*score*	*% dif*	*s dif*	*p dif*
the	168.0	1	168.0	0	168.0	1	168.0	0	0	0
the	157.0				151.0				-	-
first	129.0				123.0				-	-
question	96.5				74.0				-	-

Table 5.6: Example Repair Analysis Results for Passages 1, 2 and 3

Passage 1

When the hypotheses are only one word long the required hypothesis ("like") is second (p=2) in the hypothesis list and has the same score as the top hypothesis in the list (% dif = 0). Both when anti-grammar is used and when it is not. The anti-grammar has neither helped or hindered the system, no surprise at such a early stage in the process.

When the hypotheses are increased to two words the system difference is again 0% but the position of the required hypothesis increases seven positions when anti-grammar is used. This shows that anti-grammar is working by decreasing the likelihood of other hypotheses and leaving those hypotheses with correct grammatical structures (including the required hypothesis) with the same score.

A difference emerges when the third word is added to the hypotheses. This third word ("a") is the first word after the *interruption point* of the repair and completes the speech repair contained in the passage. The system without the

anti-grammar has the required hypothesis as eighth in the list of hypotheses and a percentage score difference of 4%, while the system with the anti-grammar has the required hypothesis 5,404th in the hypothesis list and a percentage score difference of 153% . The difference in the position of the required hypothesis between the two systems is -5,396 (i.e. the system using anti-grammar has the required hypothesis 5,396 places lower in the list than the system without anti-grammar) and the score difference is -149% (i.e. the system using anti-grammar has the required hypothesis ("like a a") 149% further away from the top hypothesis than the system without anti-grammar). This shows that the original passage contains an ungrammatical section, which is the repair, and that the inclusion of the grammatical knowledge can be seen as causing a problem or rather the repair is causing a problem to the system using grammatical knowledge.

The system without anti-grammar expands the required hypothesis beyond the repair while the system with anti-grammar stops the required hypothesis from being expanded and removes any chance of producing the required result (i.e. what the speaker actually said).

Passage 2

Passage 2 is similar to Passage 1 except that it contains a *cue phrase* ("if you like") between the repeated articles. The expansion of the hypotheses up to two words shows no differences between the results of the two systems. Both have the required hypothesis as first in their respective hypothesis lists and the same scores for the required hypotheses. The problem comes when the first word after the *interruption point* of the speech repair is added.

When the third word ("if") is added, the system without anti-grammar has the required hypothesis in fifteenth position within the hypothesis list and the percentage score difference is 26%. This drop in the position of the required hypothesis demonstrates that the system is encountering a difficulty and that there are many interpretations as to what could have been said. The system using anti-grammar has the required hypothesis 384th in the list and its percentage score difference be-

repair type	increase	no change	decrease	totals
1	17			17
2	3	1	1	5
3	6	2		8
totals	26	3	1	30

Table 5.7: Repair Type and System Performance: comparing the system with anti-grammar processing the passages containing repairs and the passages with the repairs corrected

tween the top hypothesis and the required hypothesis is 165%. The the system with anti-grammar has the required hypothesis 369 places and 138% further away from the top hypothesis. This again demonstrates the incompatibility of anti-grammar and repairs.

Passage 3

Passage 3 introduces the problem of word fragments. This example shows that the required hypothesis is never found after the location of the word fragment has been reached and that it is not possible to ignore word fragments. Word fragments are a problem that affect not only the performance of the grammar, but the system as a whole.

5.4.2 Repair vs Repaired

The previous analysis showed that an anti-grammar approach encounters difficulties when processing spontaneous passages containing speech repairs. It is not certain, however, whether these problems are caused by the anti-grammar or the repair. To check this the thirty original disfluent passages were modified to correct the actual disfluency and run through the second of the two systems (i.e. the one using anti-grammar processing).

Table 5.7 shows that of the thirty passages analysed twenty six showed an increase in performance when the repair was not present, three showed no change

Pass	Repair Passage	Repaired Passage	Diff	+/-	Rank	Signed Rank
9	—	—	0	-	1	-2
15	4	4	0	-	2	-2
23	—	—	0	-	3	-2
1	—	1	+*	+	4	12
2	—	8	+*	+	5	12
3	—	2	+*	+	6	12
7	—	9	+*	+	7	12
8	—	19	+*	+	8	12
10	—	1	+*	+	9	12
11	—	1	+*	+	10	12
13	—	1	+*	+	11	12
14	—	13	+*	+	12	12
16	—	7	+*	+	13	12
18	—	2	+*	+	14	12
20	—	6	+*	+	15	12
21	—	1	+*	+	16	12
22	—	1	+*	+	17	12
25	—	1	+*	+	18	12
27	—	8	+*	+	19	12
28	—	3	+*	+	20	12
4	5	3	2	+	21	21
12	153	145	8	+	22	22
17	12	3	9	+	23	23
26	74	5	69	+	24	24
5	156	7	149	+	25	25
19	166	10	156	+	26	26
6	193	1	192	+	27	27
24	596	19	577	+	28	28
30	1254	16	1238	+	29	29
29	54	—	-*	-	30	-30

Table 5.8: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system with anti-grammar processing the passages containing repairs and the passages with the repairs corrected

and only one showed a decrease in performance.

Table 5.8 shows for each passage (Pass), the position of the required hypothesis when the system processed the two versions of the passage (the version containing the repair (Repair Passage) and the version with the repair corrected (Repaired Passage)). It also shows the difference (Diff) between the two positions (a difference of $+^*$ shows that the exact difference can not be calculated but it is positive (i.e. the system processing the repair passage does not have the required hypothesis in the hypothesis list while the system processing the repaired passage does)), the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test[3]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system processing the repaired passage is better than the system processing the repair passage. A one-tail test is therefore appropriate.

Table 5.8 shows that the number of pairs that result in a positive outcome are 26 and the number of pairs that result in a negative outcome are 4. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system processing the repaired passage is better than the system processing the repair passage.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 4 negative outcomes, this shows that the system processing the repaired passage is significantly better (4 is $\leq$ 7) than the system processing

[3]See section 4.5, page 70, for a description of the Sign Test.

the repair passage at the p$\ll$0.005 level.

The Wilcoxon Signed Rank Test[4]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system processing the repaired passage is better than the system processing the repair passage. A one-tail test is therefore appropriate.

Table 5.8 shows the position (Rank) for each passage (Pass) when the difference (Diff) between the pairs is used to rank the passages. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system processing the repaired passage is better than the system processing the repair passage. An unknown difference, with the system processing the repaired passage not having the required hypothesis in the hypothesis list (-*), is given the maximum rank as this is an undesirable result. An unknown difference, with the system processing the repair passage not having the required hypothesis in the hypothesis list (+*), is given a minimum difference of 1 as this is a desirable result. An unknown desirable result is given a minimum difference so as not to favour the unknown difference above its minimum potential. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 429 and the total of the negative ranks is 36.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for p=0.005 is 109. As the analysis has a negative score of 36, this shows that the system processing the repaired passages is significantly better (36 is $\leq$ 109) than the system processing the repair passages at the p$\ll$0.005 level.

These tests demonstrate that anti-grammar can be of benefit to the system when the repairs are not present. Therefore, it is the repairs that are causing the problem to the system. From this we can deduce that speech repairs do cause

[4]See section 4.5, page 70, for a description of the Wilcoxon Signed Rank Test.

problems to automatic speech recognition systems using anti-grammar.

	Un-repaired passage				Repaired passage					
	like a a typical economics book				like a typical economics book				totals	
word	*top s*	*p*	*score*	*% dif*	*top s*	*p*	*score*	*% dif*	*s dif*	*p dif*
like	76.0	2	76.0	0	72.0	2	72.0	0	0	0
a	76.1	10	80.0	5	72.1	16	76.0	5	-0	-6
a	69.5	5404	176.0	153					-	-
typical					64.3	17	68.6	7	-	-
economics					60.1	12	64.2	7	-	-
book					48.7	19	53.2	9	-	-

Table 5.9: Passage 1 : Comparison of Repair & Repaired Processing

Passage Analyses

Table 5.9 (which can be compared to Passage 1 of table 5.6) shows that the processing of the modified passage allowed the hypothesis to go beyond the repair location, rather than eliminate the required hypothesis when the repair was encountered. It also shows that up to the repair location, the system performs better with anti-grammar than it does without. Comparison of the four word hypothesis "% dif" of 27% (see table 5.6) with the three word hypothesis "% dif" of 7% (see table 5.9) shows that the required hypothesis (on reaching the word "typical") is closer to the score of the top hypothesis by 20% when anti-grammar is used on the repaired passage.

Table 5.10 shows a similar result to table 5.9 in that the modified passage was expanded beyond the location of the repair. In fact the whole of the required hypothesis, which would never have been generated by the system when the repair was present, was ninth in the hypothesis list and only 4% away from the score of the top hypothesis.

The third example is not given here as once the word fragment is removed the system will be able to expand beyond the *interruption point.*

	Un-repaired passage				Repaired passage					
	describing the if you like the central				describing the central part of this course				totals	
word	*top s*	*p*	*score*	*% dif*	*top s*	*p*	*score*	*% dif*	*s dif*	*p dif*
describing	92.4	1	92.4	0	82.7	1	82.7	0	0	0
the	84.4	1	84.4	0	74.7	1	74.7	0	0	0
if	72.4	384	191.5	165					-	-
central					61.5	1	61.5	0	-	-
part					69.1	4	71.0	3	-	-
of					61.1	3	63.0	3	-	-
this					49.1	5	51.0	4	-	-
course					49.2	9	51.1	4	-	-

Table 5.10: Passage 2 : Comparison of Repair & Repaired Processing

5.5 Chapter Summary

One of the main problems with dealing with spontaneous natural speech is the number of disfluencies that appear in the speech. The results presented here identify the effect of speech repairs on the performance of a sub-section of an automatic speech recognition system using anti-grammar processing. The results demonstrate the need for a repair process, to work in collaboration with the anti-grammar, to allow the system to ignore the *reparandum* of repairs present in the passage being processed.

Table 5.4 shows that repairs within spontaneous speech do cause problems to the system. Table 5.8 shows that if these repairs don't exist in the first place (or are overcome) then the performance of the speech recognition system will increase and the anti-grammar processing will be able to perform to its potential.

Of the thirty disfluent passages analysed, 77% caused the systems performance to decrease when anti-grammar processing was included. This is un-surprising as repairs are problems which break the normal expected structures of English. As the data shows (see chapter 7) 31% of the sentences spoken during the university lecture contained disfluencies which means that 24% (77% of those sentences containing disfluencies) of all sentences spoken during the lecture would cause the system, using anti-grammar processing, to decrease in performance. This is clearly

unacceptable. A repair process must be incorporated into the system to overcome this problem.

Chapter 6

Corpora

This chapter begins with an overview of methods for the analysis of corpora and then investigates the current corpora available to researchers. Corpora are an integral part of natural language and speech recognition research. The available technology and speech recognition requirements have changed so much that the corpora themselves have had to be adapted to fit the new requirements.

6.1 Corpora Analysis

The compilation and analysis of computer corpora was generally performed by linguists [Leech and Fligelstone, 1992] whose interest lay in studying literary and linguistic texts for their own sake, without necessarily forming a clear definition of what they hoped or expected to find. Recently, however, this field has attracted strong interest from those working in information technology (e.g. speech recognition and machine translation).

Research into the structure and features of speech, as required by automatic speech recognition researchers, has been carried out in the linguistics field for many years and it would be helpful if the linguistic findings could be applied to speech recognition research. This has not been the case as the aims of linguistic research do not tie in exactly with those of speech recognition research. In their analyses

of the structure and features of speech, linguists used to remove the impurities of speech before analysing the remaining text. There motivation was in comparing types of communication rather than analysing a single communication type [Chafe and Tannen, 1987]. The speech recognition field is interested in the impurities and purities of speech thus rendering much of the linguists analyses of little relevance.

To develop solutions to the current problems being faced by automatic speech recognition systems it is necessary to undertake an analysis of relevant data. Therefore, speech analysis can be seen as an integral part of speech recognition research and speech corpora can be seen as a tool for speech researchers.

6.1.1 Text vs Speech

Comparisons of written and spoken English have been undertaken since they were identified as having separate styles. This can be traced back to Aristotle:

> *It should be observed that each kind of rhetoric has its own appropriate style. The style of prose is not that of spoken oratory.*

Chafe & Tannen (1987) provide a more detailed discussion. The findings throughout the years have been contradictory, with a claim of "*no single, absolute difference between speech and writing in English*" by Biber (1987), while Gibson *et al.* (1966) found that spoken language is more understandable, more interesting, and has a simpler vocabulary. This provided evidence for the claim by Woolbert (1922), given in [O'Donnell, 1974], that

> *Speaking and writing are alike — and different.*

For a review of the work carried out on the relationship between written and spoken English see [Chafe and Tannen, 1987] and [Liggett, 1984]. Umeda *et al.* (1992) have examined word usage and sentence structure with respect to written texts and spontaneous speech. They identify certain features which are common throughout comparisons of text and speech such as:

Different sex has no effect.
Text has a higher complexity than speech.
Speech uses easier words and simpler constructions.

[Umeda *et al.*, 1992]

All of these findings are relative to the types of texts and speech used in the analysis. Umeda *et al.* used three types of texts (literature, social science and science writing) and only one form of speech (conversation). A further finding from their analysis is that different types of text showed different uses of sentence structures. This may also be true for differing forms of speech, a single speaker oration is likely to be structurally different from a group conversation, which in turn is likely to be different from a two speaker dialogue. Therefore, to claim differences between written texts and speech is not enough, the claim should be on the differences between written texts of a specific type and speech of a specific type.

A further problem of this type of analysis is the amount of modifications made to the speech before the analysis takes place. Linguists used to remove the impurities of speech, as did Umeda *et al.* (1992), but these are an integral part of all forms of natural speech. If you remove parts of speech prior to analysis then you are not analysing speech but sections of speech, thus rendering the analysis incomplete. This does depend on the reason behind the analysis but the fundamental principle still stands. Any findings from a "text vs speech" analysis need to be carefully evaluated. The type of text and type of speech analysed, the collection of the text and speech, and the form of analysis undertaken needs to be considered. If these satisfy the requirements of the researcher then and only then may the findings be of use. If the criteria of the researcher are not met then it must be presumed that the analyses carried out are not helpful.

6.1.2 Speech vs Speech

A number of comparisons of different types of speech have been performed [Blaauw, 1992] [Daly and Zue, 1992] [Silverman *et al.*, 1992a] [Silverman *et al.*, 1992b].

Though this work is more recent than the majority of the work on the comparison of written text and speech, and is closer to the requirements of speech recognition research, the results are quite limiting.

Studies by Silverman *et al.* (1992a, 1992b) show that there are fundamental differences between read and spontaneous speech and that read speech should not be used to produce a system whose aim is to recognise spontaneous speech.

Blaauw (1992) examines the phonetic differences between read and spontaneous speech but limited her research to a single speaker, thus rendering the findings as potentially speaker dependent, rather than a global speech phenomena. Blaauw indicates that the results should be used with caution as the subject was a trained speaker and was able to perform the reading task with emphasised stress to incorporate meaning. By this statement Blaauw is indicating the possibility of sub-categories of read speech. This would be the instance of the more general hypothesis that each type of speech could have sub-categories and hence would mean that any analysis on a specific form of speech would only be relevant to research into that form of speech.

To help overcome this Daly & Zue (1992) performed their analysis on a large corpus of dialogue transactions carried out by many different speakers. However, they drastically decreased the relevance of their analysis to the field of speech recognition by the removal of disfluencies from the spontaneous speech. They then went on to compare their results with those of Koopmans-van Beinum (1990) and indicated that the fundamental reason behind the differences of the two sets of work was the initial data used. Koopmans-van Beinum used monologue speech by a speaker who was a professional radio announcer. These two pieces of work, though using the same style and process, produced radically differing results which can only be accredited to the initial data.

Therefore, it seems, to produce an automatic speech recognition system for a specific form of speech it should be developed using an analysis of the relevant speech. To produce an automatic speech recognition system for the recognition of general speech would require a full analysis of all forms of speech from pure Queen's

English to drunken speech.

6.1.3 Single Speech Analyses

Large scale analyses of single types of speech are few and far between and those that do exist examine specific aspects of the speech.

Swerts *et al.* (1992) examine the use of intonation and pauses within units of spontaneous speech. This is a prime example of speech research in which the speech used lends to the problem being assessed. A monologue of instructions is formed of clear, emphasised units. These directly assist in the identification of unit breaks.

Shriberg (1994) has carried out one of the largest analyses of speech by performing a large scale analysis of disfluencies found in dialogues. She carried out a detailed analysis looking for uniformity across different aspects such as speaker, gender and rate of speech. It is these forms of analysis that are necessary to aid automatic speech recognition research.

6.1.4 Repair Analyses

The problem being addressed in this thesis is the identification and correction of repairs. Research into this problem has been performed for a number of years (see chapter 3) but the analyses that have been carried out have mainly concerned dialogues (see table 6.1 which shows the breakdown of communication type within repair research).

It is clear that the analysis of raw speech corpora data is necessary for the progression of automatic speech recognition systems but to simply analyse speech in general or to analyse a specific type of speech to form a general theory would provide data of little use. Instead, it is necessary to analyse the specific type of speech which is relevant to the system being produced. Analyses on the data of interest to this work are limited and so it has been necessary to carry out our own analysis (see chapter 7).

Research	Data Type								
	Written	Sentences		Dialogue				Monologue	
				Human-Human		Human-Machine			
		Read	*Spon.*	*Read*	*Spon.*	*Read*	*Spon.*	*Read*	*Spon.*
Rochester					X				
SRI International						X			
SRI Cambridge						X			
INRS (Quebec)						X			
ATT & Harvard						X			
Shriberg					X	X	X		
Levelt					X				
Hindle					X				
Edinburgh					X				
Totals					5	5	1		

Table 6.1: Repair Analyses : Data Type Breakdown

6.2 Recent Corpora

> *Computer corpora are, essentially, bodies of natural language material (whole texts, samples from texts, or sometimes just unconnected sentences), which are stored in machine-readable form.*
>
> [Leech and Fligelstone, 1992, Page 115]

One of the biggest influences on the production of automatic speech recognition systems has been the types of computer corpora available to the researchers. Computer corpora (from now on know as corpora) contain high quality recorded speech, or transcribed speech, for the training and testing of speech recognisers. The data contained in early corpora consisted mainly of short utterances dealing with requests for information. This has lead to domain specific speech recognition systems with high accuracy rates. Corpora have needed to evolve and their increasing complexity, both in the domain they represent and the speaking style within the corpora, is an indication of the increasing requirements of speech recognisers and speech recognition researchers.

The problem with recent corpora (i.e. that available at the start of this research) is that the majority of the speech had been recorded in an artificial environment, which has very little background noise and disruptions to the acoustics of the signal. Systems produced and trained on data of this type generally perform well when

Corpora	Data Type								
	Written	Sentences		Dialogue				Monologue	
				Human-Human		Human-Machine			
		Read	*Spon.*	*Read*	*Spon.*	*Read*	*Spon.*	*Read*	*Spon.*
Brown	X								
LOB	X								
LLC					X				X(?)
TIMIT		X							
SCRIBE		X							
SEC					X			X	X(?)
RM				X					
Penn Treebank	X	X						X	
ATIS						X			
Map Task					X				
* WSJ								X	X
* TED					X				X
* DRA							X		
* Durham									X
Totals	3	3	0	1	4	1	1	3	5

Table 6.2: Corpora : Data Type Breakdown

tested with the speech from the corpus but this performance deteriorates rapidly when a more natural form of speech (or even data from another similar corpus) is used [Spitz, 1991]. It is therefore necessary to ensure that the data analysed is of the type that is to be recognised by the system, so that a system can be produced that will perform in the required environment. There are a large number of corpora available in many different forms (see [Edwards, 1993] for a summary). This section discusses a number of the major corpora that have been used in speech recognition research and table 6.2 shows the type of data they represent. Those marked with an '*' are the corpora that were not available at the start of this research but which have since become available.

6.2.1 Brown

The Brown corpus was compiled in the early 1960's at Brown University in the United States. It contains 500 text (written text) samples of some 2,000 words each, representing fifteen categories of American English texts printed in 1961 [Francis and Kucera, 1979]. The corpus is available in a number of versions, some with and

without part-of-speech tagging. This was one of the first machine readable corpora available and has been used in a wide variety of different analyses.

6.2.2 LOB

The Lancaster-Oslo/Bergen (LOB) corpus was compiled in the 1970s. It is a British English counterpart of the Brown corpus and contains 500 text (written text) samples of some 2,000 words each, representing fifteen categories (identical to the Brown corpus) of British English texts printed in 1961 [Johansson *et al.*, 1978] [Johansson and Hofland, 1987]. The corpus is available in a number of versions, some with and without part-of-speech tagging.

6.2.3 LLC

In 1975 the spoken English texts from the Survey of English Usage (SEU) at London were converted into a machine-readable form. Thus forming one of the first ever machine readable corpora of actual spoken English, the London-Lund Corpus (LLC). The LLC contains one hundred passages of spoken English texts of some 5,000 words each, whose origins date from 1950 [Svartvik and Quirk, 1979] [Svartvik, 1992]. The text is transcribed in considerable detail, by means of prosodic notation, showing features such as stress and intonation. The passages include seventy six recorded dialogues, six prepared monologues and eighteen unprepared monologues giving varying types of spoken English.

This corpus is helpful in that it contains a variety of different forms of communication, including prepared monologues, which is the requirement for this research. Unfortunately investigations into the transcriptions of the speech pointed to a very clean (disfluency free) form of speech which is un-typical for spontaneous speech and makes this data unhelpful for the requirements of this research.

6.2.4 TIMIT

The TIMIT corpus was developed specifically for speech recognition systems. Its main aim was to help train and evaluate speaker independent phoneme recognition systems [Lamel *et al.*, 1986]. It consists of 630 speakers (441 male and 189 female), each reading ten sentences. Though adequate for its designed task it is very limited in other domains.

6.2.5 SCRIBE

The SCRIBE corpus is a British English speech database and is similar to the TIMIT corpus in that it contains read sentences. It consists of a variety of phonetically "compact" and "rich" sentences, a two minute accent sensitive passage, ten "free" speech sentences, fifty "natural" task-specific sentences and fifty "synthetic" task-specific sentences. Seventy one speakers with four regional accents were used to record a total of over 10,000 sentences.

6.2.6 SEC

In 1984 the Lancaster/IBM Spoken English Corpus (SEC) was initiated. It contains fifty three different passages (approximately 52,000 words), representing eleven categories of contemporary spoken British English [Taylor and Knowles, 1988]. These categories range from news broadcasts to poetry and include lectures (the main interest of the work presented in this thesis). The material is available in orthographic and prosodic versions, and in two versions with part-of-speech tagging. On further analysis it was noted that the lectures were taken from radio broadcasts and Open University programmes. This suggests that they will have been highly prepared and not what can be said to be spontaneous speech. It was also noted that the transcribed speech was free from disfluencies and therefore not what was required for this research.

6.2.7 Penn Treebank

The Penn Treebank Project [Marcus *et al.*, 1993] [Marcus *et al.*, 1994] started in 1989 and has collected various pieces of text from various sources (abstracts, news stories, bulletins, book passages, Message Understanding Conference (MUC-3) sentences, manual sentences, radio transcriptions, ATIS sentences and Brown corpus passages). The aim of the project was to investigate tagging techniques. During the project over 4.5 million words of American English were (re-)tagged and some skeletal syntactic structures produced. This collection has been made available and is another source of information available to natural language researchers.

6.2.8 RM

The Resource Management (RM) corpus was one of the first major corpora produced specifically for speech recognition research and was initiated by ARPA under the Strategic Computing Speech Recognition Programme to help speech recognition researchers:

> *... design and evaluate algorithms for speaker-independent, speaker-adaptive speech, and speaker-dependent speech recognition.*
>
> [Price *et al.*, 1988]

The utterances consist of read speech taken from a human-human conversation and are made up of queries (that could form database queries) that can be linked with a naval resource management task, examples of which would be:

> **List the cruisers in the Persian sea that have casualty reports.**
> **Is the shipA's speed greater than shipB's speed.**

The utterance styles vary so that the data can be used in different development tasks and the data has been split into training and test data to allow different tasks to be performed. The corpus contains over 21,000 utterances from one hundred and sixty speakers, with varying dialects, covering approximately 1,000 different words.

6.2.9 ATIS

The Air Travel Information Service (ATIS) was the second of the corpora initiated by ARPA [Hirschman *et al.*, 1993]. It was collected by five different organisations in the United States (AT&T, BBN, Carnegie Mellon University, MIT and SRI). It contains read speech from human-machine dialogues, where the dialogues were of an air travel planning nature (i.e. a request for air travel information from a computerised database). Its multi-site collection resulted in a:

> *... wide range of variability in speaker characteristics, speech style, language style and interaction style.*
>
> [Hirschman *et al.*, 1993]

It consists of 13,975 utterances (speech and transcription) in 1,379 different scenarios (tasks) and has four hundred and seven different speakers. This corpus moves on from the speech of the RM corpus and contains more complex requests. Like the RM corpus the utterances are of a database query form, but this time they are based on air travel planning tasks, for example:

List the number of first class flights available on delta airlines.

Three types of system tests may be performed: spontaneous speech recognition tests, natural language understanding tests and spoken language understanding tests.

6.2.10 Map Task

The Map Task corpus [Anderson *et al.*, 1991] is a corpus of unscripted, task-oriented human-human dialogues which has been designed, recorded and transcribed to support the study of spontaneous speech. The scenario is that speakers must collaborate verbally to reproduce on one person's map a route which is pre-printed on the other person's map. The corpus includes both recordings and transcriptions

of the speech. A total of sixty four speakers (of which sixty one were Scottish and fifty six came from the Glasgow region) were used in one hundred and twenty eight different conversations. The conversations were controlled to allow varying levels of familiarity, eye contact and map problems. In total 146,855 words were spoken in the fifteen hours of dialogues that were recorded.

6.2.11 * WSJ

The Wall Street Journal (WSJ) corpus was under development at the start of the work presented in this thesis [Phillips *et al.*, 1992] [Bernstein and Danielson, 1992] and was therefore not available for this work. It shows the progressive development of speech corpora and speech recognition research.

This corpus was, again, initiated by ARPA and will improve upon current corpora by:

> *...focussing on the further development of speech recognition technology, toward larger or open vocabularies, speaker or task independence, and is moving towards spontaneous speech.*
>
> [Bernstein and Danielson, 1992]

Its development, like ATIS, is being undertaken by several sites in the United States. The bulk of the data is to be read news articles taken from the "Wall Street Journal" newspaper. The spontaneous speech sections are to consist of "news article type" speech where the speaker is asked to *spontaneously dictate* on a specific news topic. This moves away from the read dialogue speech of other corpora but does not quite reach the fully spontaneous monologue requirement either. The corpus is available in speech and text form and will be made up of over 400 hours of speech containing some forty seven million words.

The WSJ corpus can be used with: variable size vocabularies (5,000, 20,000 words and larger), variable perplexities (80, 120, 160, 240 and larger), speaker dependent and speaker independent training with variable amounts of data, equal

portions of verbalised and non-verbalised punctuation (to allow dictation and non-dictation applications), variable microphones and variable noise levels. Equal numbers of male and female speakers were used [Paul and Baker, 1992].

Though this corpus does contain spontaneous speech and allows researchers easy access to the recorded speech (a definite step forward in speech recognition research) it must be noted that this spontaneous speech is constrained in that it comes in small packages of utterances, is on a designated topic and was dictated in a news article style. Phillips *et al.* (1992) stated:

> *Our preliminary text processing experiment suggests that the current pre-processing scheme may not be adequate in capturing the ways people would naturally speak the sentences.*

The development of this corpus is still ongoing and it is being used in the current ARPA CSR evaluations.

6.2.12 * TED

The Translanguage English Database (TED) [Lamel *et al.*, 1994] is a corpus of recordings made of oral presentations at *Eurospeech93* in Berlin. Of the two hundred and eighty seven presentations given two hundred and twenty four were successfully recorded giving a total of seventy five hours of speech material (each talk was approximately fifteen minutes long with an extra five minutes discussion). The recordings are divided into native speakers of English and non-native speakers of English. Because of the nature of the recorded speech there will be varying dialects and accents within the speech (hence the title for the corpus). This is the type of data that would have been of particular use for the work described in this thesis. However, it is a recent development, unavailable at the current time and Lamel *et al.* (1994) only provides pre-distribution information.

6.2.13 * DRA

This corpus of human-machine dialogues was recorded using a Wizard of Oz technique [Morris, 1991] where a human (Wizard) simulates the responses of a machine while performing a dialogue with another human (Caller). A scenario was set up in which people (unaware of the nature of the experiment) could phone a route planning service to gain access to route information. The Autoroute Plus planning tool was used to obtain the information that was sent to the Callers. The voice of the Wizard was synthesised to a level where it did not resemble a human but could be easily understood. To give an idea of the synthesised voice one caller commented that "it sounds like a Dalek". The collection of the corpus was carried out in three phases. The first phase was a pilot study resulting in twenty two calls [Morris, 1991]. Twelve of these calls resulted in dialogues with the Wizard, six resulted in dialogues with a human (who intervened and took the place of the Wizard), there were three hang-ups and one incomplete call. Phase two was the first main phase which resulted in thirty one calls to the Wizard and thirteen calls to a human operator [Browning, 1993a]. Phase three involved people being asked to call the system (still unaware of the nature of the experiment) and enquire about planning routes, on the premise that "they" (the Caller) were to analyse the quality of the speech produced by the system [Browning, 1993b]. This resulted in one hundred and eighty three calls consisting of 35,123 words.

6.2.14 * Durham

The requirement for a Durham corpus resulted from the inadequacies of current corpora with respect to prepared but unscripted single speaker oration [Garigliano *et al.*, 1993]. The main bulk of the texts are undergraduate lectures which were recorded and transcribed at the University of Durham. The lectures are on a number of topics, performed by both male and female speakers whose age, experience in speaking to an audience and background all vary (although each has an academic background). The transcriptions were made as accurately as possible by including word fragments, speech disfluencies and filled pauses. Like the WSJ corpus the

Durham Corpus is still in the process of being formed. It is this corpus that has been used throughout the work presented in this thesis.

6.3 Chapter Summary

In producing a speech recognition system it is necessary to analyse the type of data that is to be dealt with by the system. To evaluate a fully functional speech recognition system the speech needs to be of the type analysed to produce the system, therefore, examples of the type of speech needs to be present within the corpora. The system at Durham requires spontaneous speech and in-particular spontaneous monologues.

The majority of the corpora available are actual transcriptions of speech or texts and do not contain the speech itself. There is no indication of what (if any) pre-processing has taken place before/during the transcription phase. The LLC, RM, ATIS, WSJ and DRA corpora are all 'speech' corpora in that the actual speech is recorded and is available as part of the corpus itself. The problem with the RM, ATIS and WSJ corpus is that they are composed of mainly read speech and, therefore, should be used to produce a system to deal with read speech. The complexity of the speech within each corpus is increasing as new corpora are being produced and, therefore, the requirement of speech recognition researchers are being met, but full spontaneous monologues have not been a necessity to date.

The corpora that contain spontaneous monologues seem to have speech which is very "clean" and therefore should be used with care. Because of the limited amount of spontaneous monologue speech in current speech corpora it was necessary to start the development of a new corpus. The Durham corpus is an attempt to fill a gap in the requirements of speech researchers by recording and transcribing large monologues (i.e. lectures and talks) so that an extra form of speech is available to speech researchers. This process is ongoing and has been a necessary requirement for the work presented in this thesis. Transcribing and annotating large bodies of speech/text is a slow and labourious process [Shriberg, 1994] but is vital if further

developments in speech recognition are to be made.

Chapter 7

Corpus Analysis

This chapter provides the details of an analysis of speech disfluencies from two sources of data. Both sets of data are introduced along with the details of the analyses carried out. The findings of the two analyses are compared with each other and with other research in the field of self-repairs.

7.1 Introduction

This analysis was carried out in two stages.

The first stage was the analysis of the Durham data and was undertaken to identify the repairs that are contained within speech relevant to this research. This involved identifying possible patterns among the repairs that do not hold for the rest of the speech. The knowledge that is identified within this analysis must be in such a form that it can be used by current automatic speech recognition systems, so that a process can be developed to identify and correct repairs. The first form of knowledge investigated was grammatical, by looking at the grammatical structure of the repairs. The second was the repair pattern, based on the work of Bear *et al.* (1992) and Shriberg *et al.* (1992). Further analyses were carried out, based on the work of Heeman (1994) and Heeman & Allen (1994a, 1994b) to determine if the *interruption point* contains enough grammatical information to solve the repair

and Hindle (1983) to examine the results of applying Hindle's 'grammatical theory' using the data identified in the analysis.

The second stage of the analysis was to generalise the results of the first analysis by examining a different style of speech (human-simulated computer) and a vast number of speakers.

7.2 The Data

Because of the limitations of corpora with respect to the information required for the work presented in this thesis and the fact that, when the required speech was available, it was not exactly clear as to what, if any, pre-processing had been carried out on the data it was necessary to develop a new corpus containing spontaneous monologues.

To generalise the findings of the first analysis an analysis of a further corpus was used for comparison purposes. This second analysis used a corpus of human-machine dialogues with many different speakers from varying backgrounds and experience.

7.2.1 The Durham Corpus

Because of the nature of developing speech corpora and the amount of time required to produce transcribed and tagged spontaneous speech to the level required, two sixty minute passages have been used, from the Durham corpus, in the analyses carried out in the work presented in this thesis (though many more have been recorded and are in the process of transcription). Each passage was fully transcribed to include all of the disfluencies of speech, including the "erm's" and "uh's" (filled pauses), word fragments and pauses, which were categorised into long and short pauses (a difficult task since humans automatically solve speech disfluencies and it was necessary to replay the speech many times in order to transcribe the speech to the level required).

The first of the two passages was used in the analysis presented in this chapter. This passage contains sixty minutes of spontaneous speech, from an experienced lecturer, on the topic of "Software Engineering". The lecture was given as part of an introduction to second year students on the practices in software engineering. This passage contained 4,903 words and 382 sentences, or part sentences, with an average of 12.8 words per-sentence.

The second of the two passages was used for black box testing of the developed system. The data was analysed but not added to the findings given in this chapter.

7.2.2 The DRA Corpus

The second corpus was used for comparative purposes to include a different speaking style into the analysis and to include a large number of speakers. This will add further evidence to the findings of the first analysis. This corpus was recorded using a "Wizard of OZ" (WOZ) technique [Morris, 1991] where a human (Wizard) simulates the responses of a machine while performing a dialogue with another human (Caller).

Of the calls recorded one hundred and seventy seven were analysed for repairs. These were where the call was actually dealt with by the Wizard and where the original recorded speech was available for analysis (without a recorded version of the dialogue it is very difficult to identify the exact location and nature of speech repairs). Of these one hundred and seventy seven calls nineteen resulted in no dialogue, leaving one hundred and fifty eight actual dialogues between a human Caller and the simulated Wizard. These one hundred and fifty eight dialogues resulted in 4,966 turns, 6,483 sentences and 35,123 words. A breakdown of the split between Caller and Wizard is given in table 7.1.

On a number of occasions the Wizard was asked to relay the directions of a requested route to the Caller. This consisted of the Wizard reciting instructions and the Caller acknowledging them. These details were not added to the original transcriptions of the dialogue and were not used in this analysis. An exception

	Caller	Wizard
Turns	2,438	2,528
Sentences	2,832	3,651
Words	11,605	23,518
Words/Sentence	4.1	6.4

Table 7.1: DRA Corpus Breakdown

was made if there was a significant interruption by the Caller which resulted in a passage of communication beyond the passing of instructions. The section of the speech corpora in which the Caller changed from speaking English to speaking French was also not included in the analysis.

The aim of including the DRA corpus into the analysis was to see if the findings of the first analysis; were specific to monologue speech or general to spontaneous speech; to determine if the findings generalised across speaking style and speaker.

7.3 The Durham Analysis

For this analysis a sixty minute passage of spontaneous speech (approximately 5,000 words), taken from the Durham corpus, was played while studying the transcription. A total of one hundred and forty five different disfluencies were identified and marked on the transcription. An analysis of these disfluencies showed that there were five disfluency types, with one type being further subdivided into four sub-groups. Though only types 1, 2 and 3 (Phonetic, Grammatical and Semantic) are of interest to us and what we would class as speech repairs (see section 2.4.1), the other classes are included for completeness.

We classify disfluencies as follows:

1. **Phonetic** - This is the repetition of a sound. The first part of the repeated sound could be a word fragment, whole word or more than one word. This is normally known as word repetition.

2. **Grammatical** - This involves a word being inserted, deleted or corrected to change the grammar of the sentence, but without changing the meaning.
3. **Semantic** - This involves a word being inserted, deleted or corrected to change the actual meaning of the utterance.
4. **Sentence Abortion** - This is the situation in which a 'clear' sentence has been aborted and another started without the original sentence being completed (i.e it is only a part sentence). Sentence abortions are normally classed as false starts and can be one of four types.
   (a) Un-required information. The details of the aborted sentence are not required within the dialogue and can therefore be ignored.
   (b) Required information. The information from the part sentence is required to be able to understand latter (probably the following) sentences.
   (c) Major repair. This is where it is possible, with major modifications, to add (join) the part sentence to the following sentence.
   (d) Completed sentence. This occurs when it is possible to add (join) the part sentence to a latter (not the following) sentence. This could be seen as sentence divergence.
5. **Dialogue repair** - This occurs when the meaning of what is said is incorrect, but the actual meaning is obvious and no repair by the speaker is made. Rather the repair is made by the listener. This could be an incorrect word, such as "your" rather than "you've", or the insertion of an un-required word in the middle of a sentence or more typically a double negative.

Disfluency types 1, 2 and 3 are normally classed as repairs and types 4 and 5 are further disfluency types which can cause problems when processing spontaneous speech. Some overlap between the disfluency types does exist, e.g. a type 3 disfluency could be classed as a type 4 disfluency if it is at the beginning of a sentence. In such cases the type 3 was prefered. The priorities used in classifying

the disfluencies were, types 1, 2 and 3 first, as they are the normal types of repairs, followed by type 4 and finally type 5.

Table 7.2 shows the relevant information on the one hundred and forty five repairs within the data. The table shows that there are twenty three sentences that contain twelve words in total. Of these twenty three sentences, six contain a single disfluency and one contains three disfluencies. Of these nine disfluencies five are of type 1, two are of type 2, one is of type 3 and one is of type 4. Of the three hundred and eighty two sentences one hundred and twenty actually contained disfluencies. This gives a high percentage (31%) of sentences in the passage containing disfluencies. Though these disfluencies are not all normally classed as repairs (only types 1, 2 and 3 are normal repairs) those that are normally classed as repairs are contained in eighty eight (23%) sentences and, following research on repairs, ninety seven (38%[1]) sentences of more than nine words long contained disfluencies. This is similar to the 34% given in [Levelt and Cutler, 1983] but greater than the 10% given in [Bear *et al.*, 1992].

Though filled pauses ("erm" or "uh") are sometimes classed as speech repairs (Heeman & Allen (1994a, 1994b) call these abridged repairs) they were not included in this analysis. The rationale being that they did not break the structure of what was said, but rather gave the speaker thinking time before continuing. The purpose of this analysis was to examine the structure of repairs. Filled pauses were very infrequent in the Durham data, in fact only thirty five were present which is low for spontaneous speech. This is probably due to the fact that the speaker is unlikely to fear interruption from the audience, therefore, there is no need to hold the conversation when thinking is required. Furthermore, the speaker is well trained and semi-prepared. Of the thirty five filled pauses only twelve appeared as part of the disfluencies and in all cases the filled pause was used as a *cue phrase* within the repair structure. The other twenty three filled pauses were simply used as thinking pauses where the speech was held up, momentarily, before continuing.

Twenty one (14%) of the one hundred and forty five repairs contained word

[1]Ninety seven of the two hundred and fifty four sentences of more than nine words.

Sentence		Disfluency frequencies				Disfluency types				
size	*Count*	*single*	*double*	*triple*	*quad*	*1*	*2*	*3*	*4*	*5*
2	7	1							1	
3	8									
4	12	2							2	
5	14	5							5	
6	17	3					2	1		
7	21	2							2	
8	23	5				2			3	
9	26	3	2			2	1	2	2	
10	26	8	1			4	2	1	3	
11	29	4	1			4			2	
12	23	6		1		5	2	1	1	
13	9	4				1	2	1		
14	19	11	1			5	1	1	4	2
15	23	4	2			2	2	1	2	1
16	21	4				1	2	1		
17	15	5	3			5	3	2		1
18	20	7				3	1	1		2
19	10	4				3		1		
20	7	2				1	1			
21	11	3			1	6		1		
22	7	3	1			3	1	1		
23	6	2	1			3				1
24	3	1	2			1	3		1	
25	5	4	1	1		3	3	3		
26	2		1				1		1	
27	2	1								1
28	4	2	1			1	1	2		
29	2	2					1			1
30	2		1			1		1		
31	1	1				1				
32	1									
33	0									
34	1									
47	1									
Total	382	99	18	2	1	57	29	21	29	9

Table 7.2: Sentence and Repair Frequencies for the Durham Data

fragments, which are a particular problem to automatic speech recognition systems. Of these twenty one fragments eleven were in type 1 repairs, six were in type 2 repairs and four were in type 3 repairs. In eighteen (86%) of the cases the word is repeated and therefore there is the possibility that this fragment could be matched in some way to the repeated word. The three non-matching word fragments were found in one type 2 repair and two type 3 repairs.

Also, three (2%) of the one hundred and forty five repairs contained *cue phrases.* This is very low but *cue phrases* need to be considered in any automatic repair process.

7.3.1 Structure Matching

The main aim of this analysis was to examine whether a pattern existed among speech repairs that is not common within the rest of speech. The first stage was to examine the grammatical structure of repairs and to compare these with the grammatical structures encountered in the rest of the data. Speech follows some form of grammar [Garigliano *et al.*, 1993] and therefore it is necessary to use a grammar in automatic speech recognition systems to help identify what is likely to have been said. The problem is that speech repairs break the standard grammatical constructs of English and are normally seen as an ungrammatical part of speech.

The analysis presented in chapter 5 shows that speech repairs are a problem for automatic speech recognition systems and cause major problems when a grammar (in this case an anti-grammar) is incorporated into the system. It is not our view that speech repairs are an ungrammatical part of speech, but rather speech repairs are an addition to the normal grammar of English. It has been stated that written text is but a sub-part of spoken text [Hindle, 1983]. Therefore, those systems that have incorporated a grammar based on the analysis of written text, which has been the mainstay of all linguistic analyses on the grammar of the English language, will not be taking these extra structures, not found in written English, into account.

Another factor is that humans can all obtain a meaning, generally the correct

Sentence	I	like	red	cars
Class	PRON	VERB	ADJ	NOUN
Structure				
1	PRON	VERB	ADJ	NOUN
2	PRON	VERB	ADJ	
3	PRON	VERB		
4	PRON			
5		VERB	ADJ	NOUN
6		VERB	ADJ	
7		VERB		
8			ADJ	NOUN
9			ADJ	
10				NOUN

Table 7.3: Example of Sentence Structures

one, from other peoples speech (i.e. people usually understand each other). Often regardless of whether we have spoken to them before and even with all of the disfluencies embedded into the speech. This suggests that disfluencies have some common nature to them. Some researchers have proposed acoustic and prosodic cues as the common element in speech repairs [Hirschberg and Nakatani, 1993] [Nakatani and Hirschberg, 1994], but here we examine the grammatical structure of the repairs in an attempt to form a grammar for these "ungrammatical" parts of spontaneous speech.

The structures for each of the one hundred and seven type 1, 2 and 3 repairs found in the Durham data were compared to all the possible combinations of grammatical structures from the whole passage. In order to achieve this each sentence in the Durham data was broken into all combinations of possible structures (using the word class e.g. VERB, NOUN, etc.) from the beginning to the end of the sentence. The sentence "I like red cars" has the structure "PRON VERB ADJ NOUN" and would, therefore, create ten different sentence structures (see table 7.3).

From the Durham data 27,268 different structures were created. One sentence structure was "VERB", which appeared 1,049 times, one was "CONJ PRON VERB", which appeared 108 times and another was "ART NOUN ART ADJ NOUN" which appeared twice.

The two lists (repair structures and sentence structures) were then compared and the frequency of the number of repair structures within the sentence structures was found. The findings were as follows:

- 76.8% of the repair structures were unique (i.e. only appeared as repairs).
- 82.7% of the repair structures that appeared in the lecture were actual repairs.
- 17.3% of those structures that were repair structures, that appeared in the lecture, were not actual repairs.

The current literature dealing with speech repairs uses a limited number of grammatical tags at specific points within a repair, but not the whole grammatical structure of the repair. The pattern of the repair structure is a more common source of knowledge that is used. These figures provide strong evidence that speech repairs have some form of, complete, grammatical structure and that these structures are uncommon amongst the rest of the data. If the repair structures were used as a grammar for repairs 17.3% of those structures investigated as a possible repair would be false positives. This may appear high but the use of extra knowledge on pattern matching should help decrease this. This technique does not rely on the matching of exact words and deals with the underlying construction of the repair, rather than the surface level words. This should ensure that the identification of repairs is not constrained to repairs within a topic. From the one hundred and seven repairs thirty four repair structures were found. These were used to form the basis for a repair grammar.

7.4 Extra Analyses on the Durham Data

A number of different researchers have identified potential solutions to the problem of speech disfluencies (see chapter 3 for current research into speech disfluencies). This section examines these theories with respect to the Durham data. The analyses given, attempt to identify the coverage of these theories for the repairs found in

the Durham data and the number of false positives that would be processed using these theories.

7.4.1 Pattern Matching

Pattern matching is the fundamental theory behind the work of Bear *et al.* (1992) and Shriberg *et al.* (1992). A repair is identified by a pattern which is created by matching words on either side of the repairs *interruption point.* The labelling system used for the patterns is given in [Bear *et al.*, 1993]. An example of this labelling system would be:

```
the green car | the red car
M1   R1  M2 | M1 R1 M2
```

See section 3.3.4 and figure 3.1 for more details of this labelling scheme.

The one hundred and seven type 1, 2 and 3 repairs within the Durham data could be described using forty one different patterns (slightly more than the thirty four grammatical structures identified in section 7.3.1). A test to identify the possible false positives was carried out by traversing the Durham data and identifying the number of times each of the forty one repair patterns were found in the data.

Of the thirty one patterns, nine contained word fragments. It is not easy to identify word fragments within a string without prior marking. Therefore, each word fragment within the data was expanded to the word it actually represented and the pattern changed from a word fragment to a full word. This reduced the number of patterns to twenty four (seven modified patterns already existed). Filled pauses were left as filled pauses within the patterns, even though they are classed as *cue phrases.* The patterns that contained filled pauses were copied and changed to remove the filled pauses, this allowed the actual repair pattern to be identified without the *cue phrase* appearing in the middle of the repair. This change increased the number of repair patterns to twenty six.

The twenty four patterns matched six hundred and thirty six sections of the Durham data while the inclusion of the two filled pause modified patterns increased the number of matches to seven hundred and thirty eight. As the twenty six repair patterns covered ninety repairs, six hundred and forty eight possible repair patterns would be actual sections of the original speech. This would give a very high false positives rate of 88% (much higher than the 17.3% of the grammatical structures). Bear *et al.* (1992) and Shriberg *et al.* (1992) used extra knowledge on parsing and acoustics to distinguish between actual repairs and false positives. The fact that the pattern matching gives a very high percentage of false positives suggests that this solution does not have a solid foundation on which to build.

7.4.2 Interruption Point

A further piece of knowledge which has been used in identifying speech repairs is the tag classes (i.e. NOUN, VERB, etc.) of the words found either side of the *interruption point* [Heeman, 1994].

The one hundred and seven repairs within the Durham data contained thirty two different two class structures around their *interruption points.* This is similar to the thirty four full repair structures. The main problem with this technique is the coverage of the 32 two class structures. In the AURAID system there are nine possible classes, so there is a total of eighty one (9x9) possible two class structures. This means that the coverage of the thirty two structures found in the one hundred and seven Durham repairs is 40% of the total possible two class structures. This is quite high when compared to the small number of repairs in the passage.

An analysis of the frequencies of these structures within the Durham data was performed. Of the thirty two structures two (6%) (covering fifteen (14%) actual repairs) are unique, in that they only appeared at the *interruption point* of a speech repair. This means that 94% of the two class structures are used in the remainder of the speech. These thirty structures appear 3,112 times, of which ninety two actually represent repairs. Therefore, using the tag classes of the words on either side of the *interruption point* would identify 3,127 possible repairs. One hundred

	Repair Types 1, 2, 3 and 4			Repair Types 1, 2 and 3		
Rule	Used	Correctly	Incorrectly	Used	Correctly	Incorrectly
1	52	51	1	50	49	1
2	7	4	3	3	2	1
3	30	19	11	25	18	7
4	4	2	2	6	4	2
Totals	93	76	17	84	73	11

Table 7.4: Results of Using Hindle's Rules on the Durham Data

and seven (3%) will be actual repairs and 3,020 (97%) would be false positives. This again is very high and seems to point to the fact that this knowledge alone is not unique to repairs.

7.4.3 Hindles Grammar Rules

Hindle (1983) looked into using a deterministic parser to identify and correct speech repairs within a string of text. Though limited in the fact that his theory used an edit signal (indicated in [Labov, 1966]), which has yet to be identified and also required word fragments and word classes to be added manually, it showed some interesting results. Hindle expanded Labov's surface level rules, designed to correct a string of words, and incorporated some syntactic constituents into the rules. A summary of the rules are given in section 3.3.3, page 34.

These are simple rules which move on from pattern matching of the words themselves to include the actual grammatical classes of the words.

For this analysis each of the repairs found in the Durham data had the rules applied. It was assumed that the edit signal existed, the tags were available, word fragments were identified and tagged as if the full word existed (i.e. word fragments were expanded into full words). As this theory was to deal with false starts as well as speech repairs, one hundred and thirty six of the speech disfluencies were used in this analysis. These included types 1, 2, 3, and 4 disfluencies.

Table 7.4 shows the results of this analysis and shows that in ninety three

(68%) of the one hundred and thirty six cases of disfluency an attempt was made to correct the problem. It also shows that in seventy six (56%) of the one hundred and thirty six cases the correction actually resulted in the required outcome. No attempt was made to correct forty three (32%) cases, which is much higher than the 2% un-attempted repairs achieved by Hindle (1983). Of the one hundred and thirty six cases seventeen (12%) were actually corrected wrongly, which is again much higher than the 3% of Hindle.

As the repairs of interest to us are of types 1, 2 and 3 the analysis for these repairs is also given in table 7.4. It shows that in 79% of the speech repairs an attempt was made to correct the problem. It also shows that in 68% of the repairs the correction actually resulted in the required outcome. No attempt was made to correct 21% of the repairs and 10% of the one hundred and seven repairs were actually corrected wrongly.

It must be noted that using rule 3 only five words before the *interruption point* were searched. This was because the majority of speech repairs contained a *reparandum* of less than five words long. Limiting the search would result in more uncorrected repairs, as those repairs with a *reparandum* of more that five words would not be processed. It would also result in less incorrect changes as the deeper the search goes the more likely a match, but not necessarily the correct match, would be found.

These are interesting results and the procedure does show some potential. However, this work relies heavily on an, as yet unknown, edit signal and so the above figures do not take into account any possible false positives that may be processed when this procedure is used with normal un-prepared sentences or even sentence hypotheses.

7.5 The DRA Analysis

The data collected at the Defence Research Agency (DRA), in Malvern, consisted of human-machine dialogues from various Callers with various speaking styles and

backgrounds. From this data one hundred and fifty eight dialogues were analysed. Table 7.5 provides the sentence breakdowns (sentence lengths) and the number of disfluencies for each sentence length. It can be seen that seventy seven (64%) of the sentences that contained disfluencies contained a single disfluency, twelve sentences contained two disfluencies, four sentences contained three disfluencies and two sentences contained four disfluencies. Only those disfluencies made by the Caller are included. Though eight disfluencies were actually made by the Wizard, these were deemed as unnatural disfluencies and were, therefore, not included. A number (22) of other disfluencies were also found which were a result of clashes between the Caller and the Wizard. These could have been included as speech disfluencies, but the analysis was aimed at self-repairs in order to allow comparisons with repairs contained in a monologue. Including disfluencies resulting from an interference, outside the speaker, was deemed unsuitable. One point of interest is that the Caller stopped immediately the Wizard started to speak in twenty one (95%) of the clashes, leaving many word fragments. In only seven (32%) of these was the word/sentence ever completed.

Again, filled pauses ("erm" or "uh") were not included as repairs in this analysis as they did not break the structure of the speech and the purpose of this analysis was to examine the structure of repairs. Though of little interest in this analysis four hundred and thirty six filled pauses, which were not part of one of the disfluencies already identified, appeared in the Caller's speech. Of these four hundred and thirty six filled pauses two hundred and seventy four (63%) occured at the start of a turn, twenty eight (7%) occured at the start of a sentence (but not at the start of a turn) and one hundred and thirty four (31%) occured within a sentence. A further twenty four filled pauses appeared within the structure of the one hundred and twenty one identified disfluencies. The frequency of filled pauses in the dialogue is much higher than the thirty five found in the Durham data. This is likely to be due to the fact that the Caller tries to keep the "floor" within the conversation, is unsure of the flow of the conversation and hence needs more thinking time than a speaker in a monologue situation. This also demonstrates that a filled pause does not necessarily result in a change in the structure of what

	Wizard	Human				
Sentence			Repairs			
length	*freq.*	*freq.*	*single*	*double*	*treble*	*quad*
1	321	794				
2	373	610	2			
3	95	365	2			
4	391	236	1			
5	376	177	4			
6	599	114	4			
7	306	110	5			
8	250	65	6	1		
9	325	70	3			
10	191	52	7			
11	125	60	9	1		
12	41	29	2	1		
13	24	29	6			
14	24	24	1			
15	34	20	4	1		
16	25	22	2	2		
17	54	9	4	1		
18	72	8	5	1		
19	15	5	1			
20	3	3	1		1	
21	1	3	2			
22	2	6		3	1	
23	1	4	1			
24	0	3	1			
25	0	1		1		
26	2	1			1	
27	1	0				
28		2	1			
29		1	1			
30		3	1			
31		0				
32		0				
33		1	1			
34		1				1
35		2				
36		0				
37		1			1	
38		0				
47		1				1
Totals	3651	2832	77(77)	12(24)	4(12)	2(8)

Table 7.5: Sentence and Repair Frequencies for the DRA Data

has been said. Only twenty four (5%) of the four hundred and sixty filled pauses in the DRA data were found to be part of a repair.

Table 7.5 also shows that the one hundred and twenty one disfluencies were contained within ninety five sentences giving an average of 3% of sentences spoken by the Caller containing repairs. This is very low compared to the 31% of the Durham analysis. The main reason for this difference is likely to be the length of the sentences within the data. The DRA data had an average of 4.1 words per sentence while the Durham data had an average sentence length of 12.8 words. Dialogues, of this nature, tend to have a large number of small sentences such as "Yes, Please", "Thank You" and "Goodbye", which dramatically decrease the average sentence length and are unlikely to contain any disfluencies. Because of this, sentences of more than nine words in length are used in analysing the number of disfluencies contained in dialogues (this is used in the work of Levelt (1983) and Bear *et al.* (1992)). In the DRA data only two hundred and ninety one (10%) of its sentences contained more than nine words of which sixty seven contained ninety two of the one hundred and twenty one disfluencies. Of the remaining 2,541 (90%) sentences twenty eight of them contained the twenty nine remaining disfluencies. This gives an average of 23% of sentences greater than nine words long containing disfluencies, which is much more comparable to the 38% of the Durham data. Table 7.6 shows figures for separate repair analyses carried out by different researchers in the fields of linguistics and automatic speech recognition. Though the percentages are quite different they all show that repairs and disfluencies are an integral part of spontaneous speech and hence it is important for automatic speech recognition systems, dealing with spontaneous speech, to deal with repairs. It also shows that the increase in human involvement results in much higher disfluency counts (Durham [Johnson *et al.*, 1994a] and Levelt [Levelt, 1983]).

Table 7.8 shows the different disfluency types found in the DRA data compared to the Durham Data. The distribution of the disfluencies is much lower for the DRA data with only five type 4 disfluencies and no type 5 disfluencies. Once again, this is likely to be due to the nature of the speech with the Callers speaking in smaller, prepared units, thus eliminating larger, more complex errors. A discussion of this

Corpus	Communication type	Frequency
Bear [Bear *et al.*, 1992]	Human-Machine (WOZ)	10%
DRA - WOZ	Human-Machine (WOZ)	23%
Durham [Johnson *et al.*, 1994a]	Monologue (Lecture)	38%
Levelt [Levelt, 1983]	Human-Human Dialogue	34%

Table 7.6: Repair Frequencies in Sentences ¿ 9 Words Long

is given in section 7.6.4.

7.6 Comparison

There are a number of ways in which repairs or disfluencies from one corpus can be compared to repairs or disfluencies from another corpus. Though different researchers give different figures within their analyses of repairs, the following are a few ways in which speech repairs and the analysis of speech repairs can be compared. The purpose of this section is to see if the findings of the analyses carried out on the Durham data and the DRA data are comparable to each other.

7.6.1 Structure

Our main area of interest in looking at repairs is their grammatical structure. Heeman & Allen (1994a, 1994b) have investigated the structure of repairs around the *interruption point* (i.e. the category of the final word of the *reparandum* and the category of the following word). They believe that the transition from one category to another, at the *interruption point* of the repair, is not likely to appear in the structure of normal speech. In their analysis editing cues or filled pauses ("uh" and "erm") are added by hand, along with word fragments, so that they can also be used in the analysis. A detailed investigation of this theory, using the Durham data has been carried out in section 7.4.2 and shows that this information alone is not enough. Rather than using the categories around the *interruption point* alone the use of the whole of the repair structure including the *correction* and the

reparandum are used here.

For this analysis a comparison was made of the structures of the repairs (disfluency types 1, 2 and 3) contained in the Durham data and those contained in the DRA data. The whole structure of the repairs was used, including word fragments (WF) and filled pauses (FP). An example of which would be:

```
 the hou- | erm the  red  house
ART WF    | FP  ART ADJ NOUN
```

Sixty one (53%) of the one hundred and sixteen DRA repairs matched a structure found in the Durham repairs. Though not conclusive this does show that there is a link between the repairs found in the two forms of communication. There are a number of possible ambiguities within the structures, such as word fragments and filled pauses, which could be removed to make the structures less specific. In order to investigate this it was decided to perform three further investigations. The first was to investigate the problem of word fragments, the second was to investigate the problem of filled pauses, while the third and final investigation was to combine the findings of the two previous investigations.

For the first investigation both sets of repairs were modified to change the word fragments so that they represented the word tag of the actual word that the speaker intended to say or more specifically the word tag of the word in the *correction* that the word fragment was supposed to match. Continuing with the previous example:

```
 the  hou-  | erm the  red  house
ART NOUN    | FP  ART ADJ NOUN
```

would result. Here the word fragment "hou-" is to match the word "house", therefore, the tag WF is changed to the tag of the matching word (i.e. NOUN). This was only done if the word fragment could be matched with a whole word in the *correction*. The reason for this was to see if the word fragment was important to the structure of the repair or whether the category that the word fragment represents was more important. Using these modified structures there was a slight increase of

three matches from sixty one to sixty four. Therefore, 55% of the one hundred and sixteen DRA structures matched the Durham structures, showing that the class appears to be more important than the fragment.

The second investigation uses the original structures again (including word fragments). These structures were modified by removing the filled pauses giving the structure:

```
 the hou- | erm the  red  house
ART WF    |     ART ADJ NOUN
```

Here the FP tag is simply removed from the structure. The reason for this was to see if the *cue phrase* is important to the identification of the repair or the structure itself is more important. Using these modified structures there was again a slight increase of eight matches taking the overall structure matches to 69 (59%). This suggests that the location of a filled pause is not primary in identifying the nature of the repair, but rather, the repair structure holds the most important information and the *cue phrase* is simply an addition to the structure, used by the speaker, to indicate a change.

The third and final investigation was to combine the two modifications used in the previous investigations. The word fragments were expanded and the filled pauses were removed from the structures of both the Durham and DRA data giving the structure:

```
 the  hou-  | erm the  red  house
ART NOUN    |     ART ADJ NOUN
```

These structures resulted in an increase of ten matches over the unmodified structures resulting in seventy one (61%) of the one hundred and sixteen DRA structures matching the Durham structures.

A summary of the results of the four investigations can be found in table 7.7. This suggests that the structure of repairs is consistent across the two sets of data and is not specific to either speaker or communication type.

Analysis	Matches
Complete Structure	61 (53%)
Removed Word Fragments	64 (55%)
Removed Editing Terms	69 (59%)
Removed Both	71 (61%)

Table 7.7: Repair Structure Matches: Comparing the DRA and Durham Repairs

From the forty five remaining structures, from the DRA data, that did not match those structures currently in the list of repair structures taken from the Durham data, nineteen additional repair structures were found. These nineteen repair structures were added to the current list of repair structures resulting in a total of fifty three repair structures. It is these fifty three generic repair structures that will be used within the AURAID system to identify and correct speech repairs.

7.6.2 Word Fragments

Word fragments are commonly found in speech disfluencies. They occur when the speaker is half way through a word and then stops to think of what to say next. The word they had started may be finished/repeated or a new word put in its place. Word fragments confuse current automatic speech recognition systems as there was little consideration into not using parts of the input signal, as they were designed to deal with read speech. It is also not practical to model all potential word fragments as this would cause dramatic increases in the size of lexicons and thus increase the search space beyond reasonable limits. Of the disfluencies found in the DRA data forty five (37%) contained word fragments which is much more than the 14% of the Durham data. Heeman & Allan (1994b) found 27% of their disfluencies contained word fragments, while Hirschberg & Nakatani (1993) found that 73% of their repairs contained word fragments. There is clearly a great diversity of values. It is unlikely that this is dependent on the nature of the speech, but rather, the experience of the speaker. The more experienced the speaker the less likely they are to include word fragments within their speech.

7.6.3 Cue Phrases

Cue phrases are another way in which disfluencies can be identified. *Cue phrases* are used by a speaker to indicate that something is wrong and that they would like to make some corrections. Words such as "oops" or "I'm sorry" can point to something being wrong and identify that a repair is taking place. Of the disfluencies in the DRA data, seven (7%) contained *cue phrases*, compared to three (2%) of the Durham data. These are low values, however, *cue phrases* do need to be taken into consideration when any solution is designed to deal with speech repairs.

Disfluency types				**Disfluency lengths**		
Type	*DRA*	*Durham*		*Length*	*DRA*	*Durham*
1	76 (63%)	57 (39%)		1	79 (65%)	70 (52%)
2	30 (25%)	29 (20%)		2	21 (17%)	26 (19%)
3	10 (8%)	21 (15%)		3	10 (8%)	6 (4%)
4a	5 (4%)	16 (11%)		4	7 (6%)	10 (7%)
4b	0 (0%)	3 (2%)		5	2 (2%)	7 (5%)
4c	0 (0%)	6 (4%)		6	2 (2%)	4 (3%)
4d	0 (0%)	4 (3%)		7	0 (0%)	2 (2%)
5	0 (0%)	9 (6%)		8	0 (0%)	6 (4%)
				11	0 (0%)	3 (2%)
				13	0 (0%)	2 (2%)

Table 7.8: Disfluency Types and Disfluency Lengths Found in the DRA and Durham Data

7.6.4 Disfluency Types

Different researchers have identified different classes of disfluency types. Heeman & Allen (1994a, 1994b) split repairs into modification repairs and abridged repairs, with false starts being removed from their data as they are too complex. Modification repairs occur when the actual text or structure of the text is changed, such as word repetitions, word insertions and word changes/modifications. Abridged repairs are where the structure does not change, but a simple *cue phrase*, editing term (such as "uh" or "erm") or word fragment is included. For the analyses presented in this thesis the disfluencies have been classed using the five category structures

given in [Johnson *et al.*, 1994a] and also summarised in section 7.3, page 131, of this thesis.

Table 7.8 shows how the disfluencies are broken down into the different categories for the DRA and Durham data. It can be seen that the DRA data contains a greater percentage of type 1 repairs and almost all (88%) are of either types 1 or 2. The Durham disfluencies are spread much more evenly amongst the different classes. This is likely to be due to the nature of the speech. The fact that the Caller is talking to a machine suggests that they are more likely to plan what they are saying, thus making fewer complex mistakes and limiting changes to repetitions or small grammatical changes. The controlled nature of the situation also limits what the Caller can say at any particular moment, thus removing complex changes. Though not proven these hypotheses suggest why disfluencies within the dialogue are of a less drastic nature than those of the monologue. Monologue speech is what can be said as unlimited and uncontrolled, except by the topic/domain of the monologue.

7.6.5 Disfluency Length

The disfluency length is the length of the text that is removed when the disfluency is completed (i.e. the length of the *reparandum* plus the *cue phrase*). So a simple two word repetition (e.g. "house | house") would have a disfluency length of one. Table 7.8 shows a comparison of disfluency lengths for the DRA and Durham data. For this analysis disfluencies of type 5 were removed from the data as identifying a disfluency length for a type 5 disfluency is not possible. This leaves one hundred and thirty six Durham disfluencies and one hundred and twenty one DRA disfluencies. It can be seen that the disfluency lengths are spread more evenly than disfluency types, but the Durham data does have much larger disfluency lengths. This is likely to be due to the more complex type 4 disfluencies found in the data. Once again the nature of the speech and the fact that the Caller will be trying not to make mistakes and to speak clearly is likely to have some effect on the sizes of the disfluencies.

7.7 Chapter Summary

The purpose of the analyses described in this chapter was to gather information on speech repairs. The first phase dealt with the form of communication relevant to the work presented in this thesis (spontaneous monologue). The second phase dealt with giving substance to the first analysis by comparing its findings with the findings of a second analysis on a different form of communication (human - machine dialogue). It was shown that there is a relationship between the findings of both analyses and that the grammatical structure of the identified repairs could be a helpful source of knowledge in overcoming speech repairs. The increase of structure matches with simple modifications showed that the inclusion of a routine for identifying word fragments and a technique for incorporating *cue phrases* would increase the coverage of the repair process. It would then be possible to produce a module that uses the grammatical structure of repairs to identify and correct speech repairs so that a satisfactory performance could be gained from an automatic speech recognition system, when dealing with spontaneous speech rather than read speech.

The analyses in this chapter identify a solution to those disfluencies that are classed as normal repairs (type 1, 2 and 3 repairs, see section 7.3). To include type 4 repairs into this solution would not be beneficial. The problem with type 4 repairs is the nature of the information that is, normally removed. In type 4 repairs and especially types b, c and d, the information held in the *reparandum* is vital to the meaning of the whole utterance. If the *reparandum* is removed it is possible that some information, on the meaning of the utterance, would be lost. To be able to incorporate type 4 repairs into any solution a much deeper semantic and possibly pragmatic analysis would have to be performed on the original utterance and the link between the two sides of the disfluency identified. Should this be completed, a sentence could be generated so that the true meaning of what the speaker intended can be identified. This is beyond the requirements of the work presented in this thesis and is in fact a research area in its own right.

There is a similar problem with type 5 disfluencies. In this case there is nothing to replace the mistake and in many cases the speaker does not seem to realise that

a mistake had been made in the first place. What we actually have is a change in the meaning of what the speaker actually wanted to say. Here pragmatic knowledge would be the only way of dealing with this problem.

Chapter 8

Solution

This chapter identifies a solution to the problem identified in chapter 2, using the knowledge gained from the analyses presented in chapter 7.

8.1 Statistical Language Model

Most work on the grammar of the English language has dealt solely with written text. This is also true for speech research related to automatic speech recognition systems. Most automatic speech recognition systems use a grammar that has been statistically created using large corpora. The data used in the production of the grammars for speech recognition systems is of a written nature and does not include the impurities of speech. It is, therefore, un-surprising that current automatic speech recognition systems can't cope with spontaneous speech. The most logical way of dealing with speech repairs, given their grammatical properties (see section 7.3.1, page 137), would be to incorporate the repairs' grammatical structures into the grammatical processing of the automatic speech recognition system. This would involve removing those anti-grammar rules, as in the AURAID system, that penalise the grammatical structures of the repairs or, as for most automatic speech recognition systems, training the system on spontaneous speech (this is the common current practice), with all the impurities of speech included. It is likely that this would help speech recognition systems produce the required

output. The recognition of a repair is, however, only a partial solution as it still needs correcting.

Humans automatically correct speech repairs, but find it difficult to read passages that contain repairs. An example[1] of such passages would be:

> In fact the the book by by Prestman was the recommended book a couple of years back.
> The third one on the list common interesting contains interesting and funny stories.
> Its like a a typical economics book.
> This book thi- this course is not abo- about the research aspects.
> The ques- the first question to answer is what is software.
> ...and this type of this is a fairly famous pie chart.
> ...and again you can notice a subtle difference between the all the three pictures now.

This difficulty in reading and understanding speech repairs, at a human level, would also arise with automatic post-processing of the output from a speech recognition system. Parsing and understanding normal text, such as that given below, is a difficult task in its own right. Adding disfluencies to the text, as in the passage above, only confuses the matter further.

> In fact the book by Prestman was the recommended book a couple of years back.
> The third one on the list contains interesting and funny stories.
> Its like a typical economics book.
> This course is not about the research aspects.
> The first question to answer is what is software.
> ...and this is a fairly famous pie chart.
> ...and again you can notice a subtle difference between the three pictures now.

[1]These examples were taken from real speech.

A human will find it easier to understand the second of the two sets of phrases, though it is possible to understand the first, and the same holds for automatic post-processing systems. For a Natural Language Processing system to process the first set of phrases it would be necessary to identify and correct the repairs. All that would be achieved by adding the structural knowledge of speech repairs to the current set up of automatic speech recognition systems would be to pass the problem onto the next level. For this reason it is desirable to detect and correct repairs before post-processing is carried out, preferably during the recognition process.

8.2 General Solution

The analyses given in chapter 7 showed that the speech repairs have common features at the grammatical level. It is these features that are going to be the basis for the solution presented in this chapter. As the solution is grammatical and problems caused by speech repairs appear when grammatical rules are used, it is necessary to perform repair processing before or at the same time as the grammatical rules are used within the system. It would be possible to deal with speech repairs as sentence hypotheses grow (i.e. each word is added to hypotheses in a hypothesis list), by identifying a repair within a hypothesis and making a copy of the hypothesis, correcting the repair and re-scoring the hypothesis at all levels. This would allow both old and new hypotheses to be present within the system, thus not eliminating the original interpretation. The difficulty with this solution is that it is unlikely that hypotheses that contain repair structures would last within the hypothesis list (i.e. it could be removed before the whole of the repair structure appears within the hypothesis). Table 5.6, page 103, shows that hypotheses containing repairs do not normally last beyond the first word of the *correction*.

The theory is to traverse the word lattice produced by the current AURAID system and identify possible repair structures. When one is identified a SKIP is added to the lattice which indicates that the system may SKIP over the *reparandum* of the repair. If the algorithm found the structure "ART NOUN ART ADJ NOUN"

in a sequence within the word lattice and certain conditions were true (i.e. the two ARTs were the same and the two NOUNs were the same (e.g. the house the red house)) then a SKIP would be added to span those phonemes that make up the first "ART" and "NOUN", allowing the sequence "SKIP ART ADJ NOUN" to be used. This solution allows both old and new word sequences to be used in the sentence hypotheses (i.e. nothing is removed from the word lattice). Therefore, it does not penalise the system's performance if the structure identified as a possible repair is not a repair. This also allows the SKIP additions to be treated as normal words and therefore limits the changes necessary to the current syntactic processing within the AURAID system. This theory could easily be adapted for use in any system which uses a word lattice structure as the basis for word level processing.

There are other additions which are required to overcome other problems which can be presented by speech repairs. The first is the presence of *cue phrases*. *Cue phrases* such as "if you like", "well" or "i mean" can appear in the middle of a speech repair. In the above example "ART NOUN ART ADJ NOUN" could easily be "ART NOUN i mean ART ADJ NOUN". To overcome this a simple change to the above algorithm can be made to allow pre-identified *cue phrases* to be present within the repair structure. Here a SKIP would be created to span the phonemes that make up "ART NOUN i mean" rather than just "ART NOUN". This again would allow the sequence "SKIP ART ADJ NOUN" to be generated.

A second addition deals with the problem of word fragments, sometimes known as part words. Word fragments cause great difficulties for speech recognisers and are fairly common in speech repairs. The only way of dealing with these is at a pre-word level. In our case we examine the phoneme transcription[2]. The phoneme transcription is traversed and sections which match later sections are identified as possible word fragments. In the phrase "the hou- the red house" the phonemes for the word fragment "hou-" and the phonemes for the beginning of the word "house" will match and therefore the possibility of there being a word fragment is identified (word fragments always occur before the repeated word and at the end

[2]As pointed out by Professor Roger Moore, in a personal communication, word fragment processing could easily be taken down to the signal processing level.

of the *reparandum*). Using this information entries representing word fragments are added to the word lattice before the repair identification is carried out. When the repair processing is undertaken word fragments are used as wild cards within the *reparandum* of the repair structure and match anything in the *correction*. The only proviso is that the relevant word within the *correction* starts at the phoneme which is identified as matching the start of the word fragment and covers all of the phonemes in the matched section of the phoneme string. This is a simple and effective solution to the problem of word fragments.

These three simple, yet effective, techniques have proved very successful in identifying and correcting speech repairs within a spontaneous speech monologue.

8.3 Detailed Solution

This section describes the detailed solution, as implemented within the AURAID speech recognition system.

There are three stages to the solution. The first deals with word fragments or part words, which are a specific problem to speech repairs. The second deals with *cue phrases*, which can appear within speech repairs. The third stage is the repair algorithm, which uses the knowledge provided by the first two stages and deals with the speech repairs themselves. Table 8.3 shows a simple example of a word lattice that could be produced from the input **Sentence** if the **Phoneme String** was corrupted to the level shown in **Corrupted String**. This example will be used throughout the explanation of the detailed solution.

8.3.1 Word Fragments

Word fragments (see section 7.6.2 for word fragment frequencies) are an integral part of speech repairs and cause a great number of problems for automatic speech recognition systems. Word fragments are not identified within automatic speech recognition systems and modelling word fragments within such systems could produce many new items and increase the search space dramatically.

A simple example of the problem of word fragments can be seen in table 8.3, this shows that the fragment "ques-" is present within the input string, but is not present within the word lattice. Thus making it impossible for any lattice parsing algorithm to trace the correct path through the lattice (the correct path is in *italics*). Jumping four phonemes would not be possible or recommended for any system. A mechanism for dealing with word fragments is required.

Class	*Name*	*Phoremes*
0	Plosive	p b t d k g
1	Affricative	tS dZ
2	Strong Fricative	s z S Z
3	Weak Fricative	f v T D h
4	Liquid/Glide	l r w j
5	Nasal	n m N
6	Vowel	i I E { A Q O U u 3 V @ aI eI oI aU @U I@ e@ U@

Table 8.1: Phoneme Classes used by AURAID

The solution presented here is a basic "pattern matching" process which spans the whole length of the signal transcription (string of phonemes) from left to right. For each phoneme (now the main phoneme) the rest of the string is checked for possible matches. A match occurs when two phonemes are equal or when two phonemes are of the same articulation class, as described in [Browning *et al.*, 1990] and shown in table 8.1. When a match is found (now the secondary phoneme) the phoneme after the main phoneme is then matched with the phoneme after the secondary phoneme. If this results in a match then the next phonemes are checked. This continues until a match is unsuccessful. A score is calculated (depending on match types, e.g. two for equality and one for articulation class) for the matched strings of phonemes. The process is then repeated. The first main phoneme is matched with the phoneme after the first secondary phoneme and then onto the phoneme after the first main phoneme. Therefore, for every phoneme in the transcription the whole of the preceeding string is checked for a match. This process is carried out for every phoneme in the string. A list containing the score, the starting position of the main matched string, the main matched string, the starting position of the secondary matched string and the secondary matched string, for every matched pair, is produced.

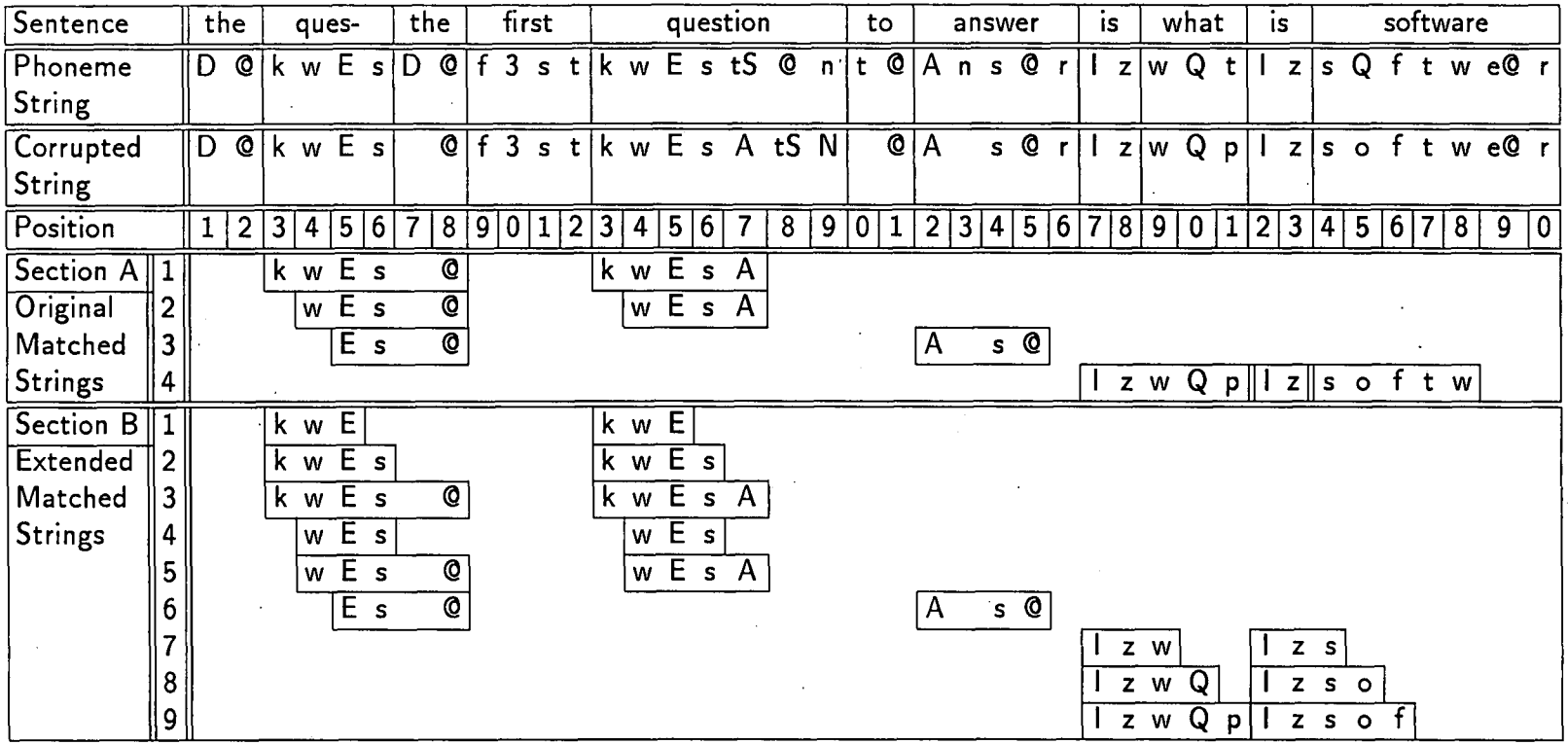


Table 8.2: Example of Phoneme Pattern Matching Results

Table 8.2 shows the sentence, the original phoneme string, the corrupted phoneme string and section A shows the position of the different matching pairs of strings for the example. The first of the pairs of matched strings represents a possible word fragment and the second is the front of the word that the word fragment matches. A number of difficulties arise with this approach. The first problem is with the final pair (4). Here the end of the main matching string (I z w Q p I z) overlaps with the start of the secondary matching string (I z s o f t w). This would not happen if the first section was a fragment of the second section. The second problem is that it may be possible for a match to go beyond the word fragment and into the following word (as occurs in the first pair (1)). In this case the first phoneme of the word after the actual word fragment is given as part of the word fragment since it matches the next phoneme in the secondary string. It may not always be necessary to expand the main string to the end of the matches.

Two simple rules can be devised, firstly to remove the overlaps and secondly to expand stretched matches, to overcome these problems. The first rule is simply to stop matching when the end phoneme of the main string is the same phoneme as the start phoneme of the secondary string. This would overcome the overlapping strings problem. The second rule would be to produce a string pair match whenever the match score is greater than a threshold (e.g. four). Using these rules the system would produce those string pairs shown in section B of table 8.2.

These nine string pairs are then used to add word fragment structures to the original word lattice. A word fragment structure is created and placed into the word lattice at the phoneme location representing the start of the main matching string. Table 8.4 shows the example word lattice with the nine fragments (WF1 to WF9) added. Also stored in this word fragment structure is the position of the secondary matched string. This is important when matching the word fragment with other words within the lattice. The word fragment can only match those words that start at the position of the first phoneme in the secondary string and must span beyond the length of the secondary string of phonemes. These word fragment structures can then be used as wild cards within any repair process, before syntactic processing takes place.

A score is associated with the identified word fragment. The score is dependent on the length of the fragment (the longer the word fragment the less likely it is to be an actual fragment) and includes penalties for both exact matching phonemes and phonemes matching within an articulation class. This score is classed as the word fragment likelihood score and is used in the same way as a word likelihood score would be used within the system.

8.3.2 Cue Phrases

Cue phrases, such as "if you like", "well" or "i mean", are a part of spontaneous speech and can appear in the middle of a repair to signal that the speaker has changed their mind or has made a mistake. An example of this could be:

the house | i mean the red house

This is not as difficult a problem as word fragments since *cue phrases* are made up of whole words. It is, therefore, possible to deal with *cue phrases* at the word lattice level, rather than having to resort to a lower level, such as phonemes. A simple solution is recommended.

A process of traversing the word lattice and identifying the location of possible *cue phrases* by following pre-defined word sequences is used. This is similar to lattice traversing, with respect to identifying the most likely spoken sentence, except that it is the existence of a particular string sequence which needs to be determined rather than the examples of *any* possible sequence. The *cue phrases* incorporated into the solution were those which existed in the Durham and DRA data and include: i beg your pardon, well, what's it, sorry, I'm sorry, no, if you like and each type of filled pause.

A 'filled pause' was added as a *cue phrase* because when ever they appeared within a repair, their action was to simulate the performance of a *cue phrase*. In fact, it is likely that a filled pause is the most definite *cue phrase* available in spontaneous speech. Filled pauses are modelled within the AURAID system,

though penalised when used. They can, therefore, be identified in the word lattice and incorporated into a repair structure.

A list of all of the *cue phrase* structures found in the word lattice is kept. A score is associated with the identified *cue phrase* and includes the scores of each of the words that constitute the *cue phrase* (so that spanning the length of the *cue phrase* is penalised to the same level as the words themselves). In addition there is a *cue phrase* penalty score, which allows the actual words to have preference over the *cue phrase.* It is only necessary to keep the start and end locations (phonemes) of the *cue phrase* and the score associated with the *cue phrase.* This information can then be used within the repair process to allow *cue phrases* to appear between the *reparandum* and *correction* of a repair. The word lattice given in table 8.3 shows that, from the *cue phrases* found in the data and used within the AURAID system, there is only one possible *cue phrase* and that is the word "well" found in the latter section of line one.

8.3.3 Repair Analysis

The above two processes produce information that can be used by a repair process. The first piece of information contains all of the possible word fragments, with associated scores, while the second contains all of the possible *cue phrases* and their associated scores.

The algorithm suggested for tackling speech repairs relies on the grammatical structure of repairs and the knowledge collected by the word fragment and *cue phrase* processes. This knowledge is used to traverse the word lattice (produced within the AURAID system) to identify repair structures. If a repair structure is found within the lattice a **conditional** SKIP is added to the lattice. This SKIP allows a subsequent lattice parser to span (ignore) those words identified as being the *reparandum.* The SKIP is **conditional** in that it should only be used if followed by those words that were identified as the *correction* of the repair.

Sentence		the	ques-	the	first	question	to	answer	is	what	is	software
Phoneme String		D @	k w E s	D @	f 3 s t	k w E s tS @ n	t @	A n s @ r	I z	w Q t	I z	s Q f t w e@ r
Corrupted String		D @	k w E s	@	f 3 s t	k w E s A tS N	@	A s @ r	I z	w Q p	I z	s o f t w e@ r
Position		1 2	3 4 5 6	7 8	9 0 1 2	3 4 5 6 7 8 9	0 1 2 3 4 5 6	7 8	9 0 1 2 3	4 5 6 7 8 9 0		
Word Lattice	1	*the* (ART)	question (NOUN)		quest (NOUN)	on (PREP)		are (VERB)	*is* (VERB)	well (ADJ)	soft (ADJ)	wear (VERB)
	2	there (ADV)		thirst (VERB)		ton (NOUN)	dance (NOUN)		out (ADV)		offer (VERB)	
	3	they (PRON)		*the* (ART)	fist (NOUN)	*question* (NOUN)	tan (NOUN)	quiz (NOUN)	*what* (ADJ)	ice (NOUN)	safe (NOUN)	
	4	how (ADV)		turf (NOUN)	square (ADJ)	ton (NOUN)	*to* (PREP)	*answer* (VERB)	ice (NOUN)	which (PRON)	*software* (NOUN)	
	5	their (ADJ)		*first* (ADJ)			town (NOUN)		*is* (VERB)	swiftly (ADV)	ware (NOUN)	
	6	the (ADV)	crest (NOUN)	fresh (ADJ)		nine (ADJ)		switch (NOUN)		safty (NOUN)		
	7	they (PRON)		tease (VERB)	channel (NOUN)	and (CONJ)		twist (VERB)		ear (NOUN)		

Table 8.3: Example Word Lattice Used Throughout the Solution Explanation

Sentence	the	ques-	the	first	question	to	answer	is	what	is	software
Phoneme String	D @	k w E s	D @	f 3 s t	k w E s tS @ n	t @	A n s @ r	I z	w Q t	I z	s Q f t w e@ r
Corrupted String	D @	k w E s	@	f 3 s t	k w E s A tS N	@	A s @ r	I z	w Q p	I z	s o f t w e@ r
Position	1 2	3 4 5 6	7 8	9 0 1 2	3 4 5 6 7 8 9	0 1	2 3 4 5 6	7 8	9 0 1	2 3	4 5 6 7 8 9 0
Word Lattice 1	*the* (ART)	question (NOUN)			quest (NOUN)	on (PREP)		are (VERB)	*is* (VERB)	well (ADJ)	soft (ADJ)	wear (VERB)
2	there (ADV)	WF1 (13-15)	WF6 (22-25)	thirst (VERB)		ton (NOUN)	dance (NOUN)		WF7 (32-34)	out (ADV)		offer (VERB)
3	they (PRON)	*WF2* (13-16)	*the* (ART)	fist (NOUN)	*question* (NOUN)		tan (NOUN)	quiz (NOUN)	*what* (ADJ)	ice (NOUN)	safe (NOUN)	
4	how (ADV)	WF4 (14-16)	turf (NOUN)		square (ADJ)	ton (NOUN)	*to* (PREP)	*answer* (VERB)	ice (NOUN)	which (PRON)	*software* (NOUN)	
5	their (ADJ)	WF3 (13-17)		*first* (ADJ)			town (NOUN)		WF9 (32-36)	*is* (VERB)	swiftly (ADV)	ware (NOUN)
6	the (ADV)	crest (NOUN)		fresh (ADJ)		nine (ADJ)			switch (NOUN)		safty (NOUN)	
7	they (PRON)	WF5 (13-17)			tease (VERB)	channel (NOUN)	and (CONJ)		WF8 (32-35)	twist (VERB)		ear (NOUN)

Table 8.4: Example Word Lattice with Added Word Fragments

Sentence		the	ques-	the	first	question	to	answer	is	what	is	software
Phoneme String		D @	k w E s	D @	f 3 s t	k w E s tS @ n	t @	A n s @ r	I z	w Q t	I z	s Q f t w e@ r
Corrupted String		D @	k w E s	@	f 3 s t	k w E s A tS N	@	A s @ r	I z	w Q p	I z	s o f t w e@ r
Position		1 2	3 4 5 6	7 8	9 0 1 2	3 4 5 6 7 8 9	0 1	2 3 4 5 6	7 8	9 0 1	2 3	4 5 6 7 8 9 0
Word Lattice	1	*the* (ART)	question (NOUN)		quest (NOUN)	on (PREP)		are (VERB)	*is* (VERB)	well (ADJ)	soft (ADJ)	wear (VERB)
	2	there (ADV)		thirst (VERB)		ton (NOUN)	dance (NOUN)		out (ADV)		offer (VERB)	
	3	they (PRON)		*the* (ART)	fist (NOUN)	*question* (NOUN)	tan (NOUN)	quiz (NOUN)	*what* (ADJ)	ice (NOUN)	safe (NOUN)	
	4	how (ADV)		turf (NOUN)	square (ADJ)	ton (NOUN)	*to* (PREP)	*answer* (VERB)	ice (NOUN)	which (PRON)	*software* (NOUN)	
	5	their (ADJ)		*first* (ADJ)			town (NOUN)		*is* (VERB)	swiftly (ADV)	ware (NOUN)	
	6	the (ADV)	crest (NOUN)	fresh (ADJ)		nine (ADJ)		switch (NOUN)		safty (NOUN)		
	7	they (PRON)		tease (VERB)	channel (NOUN)	and (CONJ)		twist (VERB)		ear (NOUN)		
Added Skips	8	SKIP (the PW2)		the ART	— ADJ	quest NOUN						
	9	SKIP (the PW2)		the ART	— ADJ	question NOUN						
	10	SKIP (the question)			— ADJ	question NOUN						

Table 8.5: Example Word Lattice after Repair Processing has been Performed

This technique allows both the original word sequence and the repaired sequence to be used. Nothing is removed from the search, but extra knowledge is used to ensure that every possibility is explored before a final decision is made.

Repair Structures

The algorithm is based on the analyses of speech repairs carried out on the Durham and DRA data. These analyses were undertaken specifically for the work presented in this thesis and have proven invaluable in identifying a solution to the problem of speech repairs.

From the one hundred and seven repairs found in the Durham data thirty four different repair structures were identified (embedded repairs were split into separate structures) and shown to be uncommon to the structures of normal speech. In most cases a repair structure only appeared as a repair and not in any other part of the passage. A comparison was made between the repairs found in the Durham data and the repairs found in the DRA data. This provided evidence that repair structures are common across speaking styles and speakers. Those DRA repairs that did not match one of the structures of the Durham repairs resulted in nineteen new structures making a total of fifty three different repair structures.

It is these fifty three grammatical structures that are the basis for identifying and correcting speech repairs. A typical grammatical structure for a repair would be:

```
  the     house    |  the     red     house
ART(1) NOUN(1)     | ART(2) ADJ NOUN(2)
```

Here "ART(1) NOUN(1)" is the *reparandum* and "ART(2) ADJ NOUN(2)" is the *correction*. This in its own right is not classed as a repair structure, as certain other conditions must apply to make the structure valid. "ART(1)" must represent the same word as "ART(2)" and "NOUN(1)" must represent the same word as "NOUN(2)", as in the above example. These conditions are important to eliminate false positives on structures that could easily appear within a large word lattice.

From now on two classes which match at the word level will also be given the same suffix number, therefore, the above structure will look like:

```
the     house   | the   red   house
ART(1) NOUN(2) | ART(1) ADJ NOUN(2)
```

This notation distinguishes between the same class in the same section and shows which classes match. Each repair structure was given a score. This score was based on the classification of the repair, using the classification scheme developed for this work, and also the frequencies of the repair structures within the analysed data. The purpose of the score is to ensure that when a repair structure is used it does not have preference over the normal sequence of words unless other modules, such as anti-grammar, penalise the original sequence.

Another structure which is added to these repair structures are repeated words (where the speaker simply repeated what they previously said). This is the simplest form of speech repair and is the most common. Those repair structures that dealt with repetitions were removed from the repair structures and word repetitions were dealt with in the same way as repair structures, except the words themselves were taken into account. Repeated patterns of one, two or three words in length were searched for and dealt with in the same way as a repair structure, by adding a conditional SKIP over the *reparandum.*

Lattice Parsing

A word lattice was deemed as the most appropriate level to perform speech repair processing. This is based on three reasons:

- a grammar, which causes problems, has yet to be applied.
- the word lattice level is the first level in which the grammatical knowledge is present within the AURAID system.
- limited modifications to the actual hypothesis generation section of the system would be required.

The lattice parsing algorithm needs to determine whether, a) one or more of the repair structures could start from a particular point or b) a repeated word list of one, two or three words in length can be spanned. If either of these are possible then a SKIP is added to the lattice.

There are three phases to parsing a lattice when looking for repair structures. The first is to identify the *reparandum* and store any words that require a matching word in the *correction.* The second phase is to identify the *correction.* This second phase needs to be repeated.

The first search of the second phase starts directly after the *reparandum.* A *correction* is only valid if all classes match and all words match with the *reparandum*, where necessary. For the repair structure "ART(1) NOUN(2) | ART(1) ADJ NOUN(2)" the following examples are true:

the house | the red house = match

the house | the red car = no match

the house | the house house = no match

the house | the brick house = match

The second search of the second phase is to check to see if any *cue phrases* start where the *reparandum* ends. If so, a search for a *correction* is started at the position where the matched *cue phrase* ends. This allows a *cue phrase* to be present between the *reparandum* and *correction* of a repair structure.

The third and final phase is to create a SKIP entry in the word lattice. Once a *reparandum* and *correction* is found then a SKIP can be produced. The SKIP is added to the lattice to start where the first word of the *reparandum* starts and end where the last word of the *reparandum*, or *cue phrase* if applicable, ends. This SKIP is given a score which is comprised of the sum of the scores of the words which the SKIP is attempting to replace, a pre-calculated repair score and the *cue phrase* score, if applicable. The word scores are used so the likelihood of the SKIP being present is reflected by the likelihood of the words being present. The pre-calculated repair score is added so that the actual sequence of words the SKIP is

attempting to replace has preference (where necessary) over the SKIP itself. This ensures the importance of grammatical knowledge in positioning hypotheses in the list is retained. The *cue phrase* score is added to take the liklihood of the *cue phrase* into account.

Because of the nature of the knowledge used in identifying speech repairs any SKIP placed in a word lattice must be conditional. By conditional we mean that it can only be followed by those words which were identified as being the *correction* of the words the SKIP is attempting to replace (i.e. the *reparandum*). Therefore, along with a SKIP score it is necessary to store information on the following words (*correction*).

Table 8.6 shows the progression of the repair algorithm at each stage using two repair structures ("ART(1) NOUN(2) | ART(1) ADJ NOUN(2)" and "ADJ NOUN(1) | ADJ NOUN(1)") and the word lattice given in table 8.3. Example 1 shows that, the fact that, the first NOUN is a word fragment is causing problems for the repair process (as it does for the normal hypothesis generation process). It shows that there are three possible *reparandum* that are made up of "ART NOUN". However, no structures follow that have the structure "ART ADJ NOUN" and satisfy the match criteria (i.e. matching ARTs and matching NOUNs). Example 2 shows that although there are fourteen occurrences of the *reparandum* within the lattice only four of them (6, 8, 9 and 14) are followed by the appropriate *correction* structures. Only two of these *correction* structures satisfy the matching criteria (as shown by "Y" after the *correction* process within table 8.6) and resulted in SKIP structures. The first SKIP structure covers phonemes 1 to 8 and must be followed by the words "first(ADJ) question(NOUN)". The second SKIP structure covers phonemes 1 to 8 and must be followed by the words "fresh(ADJ) question(NOUN)".

Example 1	Example 2
ART(1) NOUN(2) — ART(1) ADJ NOUN(2)	ADJ NOUN(1) — ADJ NOUN(1)
phase 1 - identify the *reparandum*	phase 1 - identify the *reparandum*
1) the(line 1, position1) question(1,3) 2) the(1,1) crest(6,3) 3) the(3,7) fist(3,9)	1) well(1,29) safe(3,34) 2) well(1,29) software(4,34) 3) well(1,29) safty(6,34) 4) soft(1,34) ware(5,38) 5) soft(1,34) ear(7,38) 6) what(3,29) ice(3,42) 7) square(4,11) tom(4,18) 8) their(5,1) question(1,3) 9) their(5,1) crest(5,3) 10) first(5,9) quest(1,13) 11) first(5,9) quesion(3,13) 12) fresh(5,9) quest(1,13) 13) fresh(5,9) quesion(3,13) 14) nine(6,29) quiz(3,24)
phase 2 - identify the *correction*	phase 2 - identify the *correction*
1) no following structure 2) no following structure 3) no following structure	1) no following structure 2) no following structure 3) no following structure 4) no following structure 5) no following structure 6) soft(1,34) ware(5,38) = N 6) soft(1,34) ear(7,38) = N 7) no following structure 8) first(5,9) quest(1,13) = N 8) first(5,9) question(3,13) = Y 8) fresh(5,9) quest(1,13) = N 8) fresh(5,9) question(3,13) = Y 9) first(5,9) quest(1,13) = N 9) first(5,9) question(3,13) = N 9) fresh(5,9) quest(1,13) = N 9) fresh(5,9) question(3,13) = N 10) no following structure 11) no following structure 12) no following structure 13) no following structure 14) well(1,29) safe(3,34) = N 14) well(1,29) software(4,34) = N 14) well(1,29) safty(6,34) = N 14) what(3,29) ice(3,42) = N
phase 3 - create skips	phase 3 - create skips
No matches therefore no skips to create	skip phonemes 1-8 followed by (first question) skip phonemes 1-8 followed by (fresh question)

Table 8.6: Example Repair Process Using the Normal Word Lattice (table 8.3)

8.3.4 Repair Analysis Extension Using Word Fragment and Cue Phrase Knowledge

One problem not addressed by the above algorithm is the presence of word fragments. Possible word fragments (see section 8.3.1), are placed into the word lattice as wild cards. These can be used at the end of a *reparandum* to match any word which starts at the phoneme position of the secondary phoneme string (given in the matched pair of phoneme strings), which matched the word fragments phoneme string and spans beyond the final phoneme position of the secondary phoneme string. Table 8.4 shows that WF1 (word fragment 1 (line 2, position3)) will match any word which starts at phoneme 13 (k) and spans beyond phoneme 15 (E). So, if the repair structure was "ART(1) NOUN(2) | ART(1) ADJ NOUN(2)" and the *reparandum* was "the WF1" then the end of the "ADJ" word should be at phoneme 12, the start of the *correction* "NOUN(2)" should be at phoneme 13 and the end of the *correction* "NOUN(2)" should be at least phoneme 16. If these conditions apply then the structure found could be a repair, where the first occurrence of the "NOUN(2)" is a fragment of the second occurences. An example of this would be "the hou- | the red house".

Table 8.7 shows the additional findings (i.e. it does not show those given in table 8.6) at each stage within the repair process, using the same two repair structures ("ART(1) NOUN(2) | ART(1) ADJ NOUN(2)" and "ADJ NOUN(1) | ADJ NOUN(1)"), when the extra knowledge of word fragments is added to the word lattice (i.e. using table 8.4) and the knowledge on *cue phrases* is used (one possible *cue phrase* of 'well' at position 29 in the phoneme string (see section 8.3.2)).

Example 1, of table 8.7, shows that, with the addition of word fragments (WF1, WF2, etc. in table 8.4), SKIPs can be produced that span the section of the phoneme string that needs to be removed (phonemes 1 to 6). More than one SKIP is produced (see phase three of table 8.7) as the word fragments are wild cards and match any word. It can be seen that the required repair structure is the second SKIP produced in example 1 (i.e. SKIP phonemes 1 to 6 with a *correction* of the(ART) first(ADJ) question(NOUN)). There are other SKIPs produced and all

will be added to the word lattice. The hypothesis generation process is able to choose which, if any, of the SKIPs are used. The incorporation of these SKIPs into the lattice means that the correct path is available for the hypothesis generation process. The repair process provides additional options to the system. The key element of the process is that the *reparandum* section of the phoneme string can be spanned by the system, if required.

Example 1	Example 2
ART(1) NOUN(2) — ART(1) ADJ NOUN(2)	ADJ NOUN(1) — ADJ NOUN(1)
phase 1 - identify the *reparandum*	phase 1 - identify the *reparandum*
1) the(line 1, position1) WF1(2,3) 2) the(1,1) WF2(3,3) 3) the(1,1) WF3(5,3)	1) their(5,1) WF1(2,3) 2) their(5,1) WF2(3,3) 3) their(5,1) WF3(5,3)
phase 2 - identify the *correction*	phase 2 - identify the *correction*
1) no following structure 2) the(3,7) first(5,9) quest(1,13) = Y 2) the(3,7) first(5,9) question(1,13) = Y 2) the(3,7) fresh(5,9) quest(1,13) = Y 2) the(3,7) fresh(5,9) question(1,13) = Y 3) no following structure	1) no following structure 2) no following structure 3) first(5,9) quest(1,13) = N 3) first(5,9) question(1,13) = Y 3) fresh(5,9) quest(1,13) = N 3) fresh(5,9) question(1,13) = Y
phase 3 - create skips	phase 3 - create skips
skip phonemes 1-6 followed by (the first quest) skip phonemes 1-6 followed by (the first question) skip phonemes 1-6 followed by (the fresh quest) skip phonemes 1-6 followed by (the fresh question)	skip phonemes 1-8 followed by (first question) skip phonemes 1-8 followed by (fresh question)

Table 8.7: Example Repair Process using Word Fragment Information (table 8.4)

Example 2 in table 8.7 demonstrates how the inclusion of word fragments can allow even more possibilities. These possibilities need to be examined so that the hypothesis generation process can identify the most appropriate path through the lattice. It must be noted that a full sequence of words has preference over a sequence incorporating SKIPs as there are penalties, above the normal word scores, added to all SKIPs. Additional knowledge on grammar and semantics can then re-score hypotheses allowing repaired sections to be selected, if they are valid options.

In example 2 of table 8.7 the introduction of word fragments has resulted in two extra repair structures which have the same structure as those identified using no word fragments (see table 8.6). On such occasions those repair structures which do not use the word fragment information are created while the others are not (they are more specific, in that they identify the *reparandum* in detail). This would leave 6 (see table 8.1) of the 8 repair structures identified in tables 8.6 and 8.7.

SKIP	the first quest	
(the WF2)	(ART) (ADJ) (NOUN)	
SKIP	the first question	
(the WF2)	(ART) (ADJ) (NOUN)	
SKIP	the fresh quest	
(the WF2)	(ART) (ADJ) (NOUN)	
SKIP	the fresh question	
(the WF2)	(ART) (ADJ) (NOUN)	
SKIP	first question	
(the question)	(ADJ) (NOUN)	
SKIP	fresh question	
(the question)	(ADJ) (NOUN)	

Figure 8.1: SKIPs found in table 8.4

An additional technique for decreasing the search space can be applied. This involves joining those SKIPs that have similar structures, in that the only difference is with the none matched section within the *correction*. For example, in the structure "ART(1) NOUN(2) | ART(1) ADJ NOUN(2)" the adjective that appears within the *correction* is not dependent on any other part of the structure. The only thing that does effect it is the start position of the following word and the end position of the previous word. Those words within the *correction* that are not specifically dependent on a word within the *reparandum* can be replaced by a wild card, matching only the word's class. This allows all those repairs identified to be merged if the ADJs (both examples have un-matching ADJs) occupy the same

space within the phoneme string. This is true for all structures, in both examples, therefore, the six structures shown in figure 8.1 can be reduced to just three repair structures which are shown in figure 8.2. These structures can be seen in table 8.5 where the SKIP information has been added to the word lattice.

SKIP	the	—	quest
(the WF2)	(ART)	(ADJ)	(NOUN)
SKIP	the	—	question
(the WF2)	(ART)	(ADJ)	(NOUN)
SKIP	—	question	
(the question)	(ADJ)	(NOUN)	

Figure 8.2: General SKIPs found in table 8.4

This additional technique will require an extra change to the hypothesis generation process so that when a wild card structure is found within the *correction* (i.e. following words) all those words of the appropriate size and class can be added to the hypothesis. This is not essential for the algorithm to work, but is useful in reducing the search space within the system.

8.3.5 Repair Analysis Extension (Embedded Repairs)

Another problem, which is not addressed by the above algorithm, is the problem of embedded repairs. The data shows that it is possible that one repair can be embedded within another repair. Consider the following example:

the house | the | the red house.

Here there are two *interruption points.* The first indicates the repair "the house | the red house" while the second indicates the repair "the | the". During the lattice parsing process the first repair would never be identified as the *reparandum* "the

house" is not followed by the *correction* "the red house". Therefore, the first repair would not be corrected. To overcome this a two pass strategy is adopted in which the second pass searches for speech repairs that could contain repairs created on the first pass. In the above example the first pass would create a SKIP for "the | the" while the second pass would use the SKIP of the first pass to span the unrequired word ("the"), thus completing the connection between the *reparandum* and *correction*. The original SKIP would still be present within the lattice and would be available, if required, during the hypothesis generation process.

8.3.6 Repair Analysis Extension (Filled Pauses)

Filled pauses can cause difficulties for automatic speech recognition systems. In this work they are not distinguished as specific repairs, but are acknowledged as a problem. The AURAID system models filled pauses within its lexicon, but penalises their use. They are, therefore, rarely used in the final result of the system. We have incorporated filled pauses into our repair process by adding a structure to the list of repair structures to SKIP an interjection (the class used for a filled pause within the lexicon) and allow it to be followed by a word of any other class. This will create a SKIP within the word lattice whenever a filled pause is identified as a possible word. This allows the hypothesis generation process to ignore locations identified as filled pauses.

8.3.7 Repair Analysis Conclusion

The repair process, described above, generates the necessary information for the hypothesis generation process to span the *reparandum* of a repair. The algorithm described identifies the repair by using the repair structures produced from the data analysis process. The information generated includes:

- the length of phonemes that can be omitted including the *reparandum* and *cue. phrase.*

- the words that must follow the spanned length of phonemes.

A two pass strategy allows embedded repairs to be covered and the use of knowledge produced by the phoneme matching algorithm allows word fragment information to be used within the repair process. Table 8.5 shows an example word lattice once word fragment, *cue phrase* and repair processing have been performed.

After the repair processing is complete and all of the SKIPs have been added to the word lattice, the word fragments need to be removed. This is necessary since if they were retained in the lattice they would be used by the hypothesis generation process, as a normal word. This would allow the word fragments to be part of any hypothesis without taking the matching phoneme string into account.

8.3.8 Hypothesis Generation

The hypothesis generation process is that part of the AURAID system that identifies possible sentences by traversing the word lattice. Table 8.8 shows the first four cycles in the production of possible sentences using the original word lattice given in table 8.3. The first cycle creates a hypothesis for each word that starts at the first phoneme[3]. During the second cycle a word is added to the end of each of the hypotheses produced by the first cycle. Table 8.8 shows that in cycle two it is possible to add two words (question and crest) to the end of the sentence hypothesis 'the'. Only those words which start at the phoneme after the final phoneme of the final word in the hypothesis will be added. Those hypotheses which have a '*' attached can not be expanded as no word follows them. They are then removed from the hypothesis list as, when using them, it is not possible to span the whole length of the lattice. The cycles continue until the end of the lattice is reached and the best path through the lattice is taken as the actual sentence spoken. At each stage each hypothesis is given a score and the hypotheses are sorted based on this score. It is during this scoring process that the problem with processing repairs

[3]There are two hypotheses for the word "the" one for the ART and one for the ADV. There are two hypotheses for the word "they" one which covers three phonemes and one which covers four phonemes.

appears. The grammar module is penalising those hypotheses that contain repairs.

This example (table 8.8) also shows the problems posed by word fragments. As word fragments are not a part of the word lattice, it is not possible for the system to follow the required path through the lattice. So on cycle 3 the required hypothesis 'the ques- the' is not present.

Cycle 1	Cycle 2	Cycle 3	Cycle 4
the	the question	the question fist	the question fist quest
there	the crest	the question first	the question fist question
they	there question	the question fresh	the question first quest
how	there crest	the crest fist	the question first question
their	they question	the crest first	the question fresh quest
the	they crest	the crest fresh	the question fresh question
they	how *	there question fist	the crest fist quest
	their question	there question first	the crest fist question
	their crest	there question fresh	the crest first quest
	the question	there crest fist	the crest first question
	the crest	there crest first	the crest fresh quest
	they *	there crest fresh	the crest fresh question
		they question fist	there question fist quest
		they question first	there question fist question
		they question fresh	...
		they crest fist	...
		they crest first	...
		they crest fresh	...
		their question fist	...
		their question first	...
		their question fresh	...
		their crest fist	...
		their crest first	the question frest quest
		their crest fresh	the question frest question
		the question fist	the crest fist quest
		the question first	the crest fist question
		the question fresh	the crest first quest
		the crest fist	the crest first question
		the crest first	the crest fresh quest
		the crest fresh	the crest fresh question

Table 8.8: Example Lattice Parsing Run Using the Normal Word Lattice (table 8.3). (Shows the hypothesis list at the end of each cycle for the first four cycles of the Lattice Parsing algorithm.)

There are a limited number of changes required to the hypothesis generation process to incorporate SKIPs, placed into the lattice, into the repair process. The first is that conditional SKIPs must be propagated. A SKIP can appear at any stage within a hypothesis, but when a SKIP is at the end of a hypothesis, the

words indicated as following (i.e. the *correction*) must come next. The hypotheses increase by one word at a time and generally these words are taken from the word lattice. When a SKIP is used, a list of words that must follow the SKIP is retained. When the next word is added to the hypothesis containing the SKIP, it is taken from the front of the stored list (and not from the lattice) and that word is then removed from this list. Once this list is empty the hypothesis can be expanded using any of the appropriate words in the word lattice. Figure 8.3 shows a typical sequence of cycles on a single hypothesis[4]. The list of words in brackets is the stored list and appears when a SKIP is used. It decreases as each word is added to the hypothesis. After cycle six the SKIP has been processed and any appropriate word, taken from the lattice, can now be added to the hypothesis.

Sentence = I like the house | the red house on the corner
Repair = (SKIP) | (the red house)

Cycle 1 = I

Cycle 2 = I like

Cycle 3 = I like SKIP (the red house)

Cycle 4 = I like SKIP the (red house)

Cycle 5 = I like SKIP the red (house)

Cycle 6 = I like SKIP the red house

Cycle 7 = I like SKIP the red house on

Cycle 8 = I like SKIP the red house on the

Cycle 9 = I like SKIP the red house on the corner

Figure 8.3: Example Hypothesis Generation with SKIPs

The second modification to the hypothesis generation process deals with wild cards within the repair structures. This modification is only necessary if repair entries are combined and non-matching words within the *correction* are stored as

[4]At each stage when a word is taken from the lattice the number of hypothesis will increase as more that one word could be added to the hypothesis. This is not shown here.

wild cards. In such cases the list of words to follow, within the hypothesis, will contain a wild card entry. This wild card entry indicates the class of the word (e.g. NOUN) and the length of the word in phonemes. The hypothesis generation process must find all words, within the lattice, that start directly after the end position of the hypothesis, that match the necessary word class and end at the final phoneme location identified by the wild card. Each successful word can then be used to create a new hypothesis with that word being the final word in the hypothesis and the list of words to follow within the hypothesis being decreased by one.

```
Sentence = I like the house | the red house I mean | red car
Repair 1 =        (SKIP    ) | (the red house)
Repair 2 =                          (SKIP          ) | (red car)

Cycle 1 = I

Cycle 2 = I like

Cycle 3 = I like SKIP (the red house)

Cycle 4 = I like SKIP the (red house)

Cycle 5 = I like SKIP the red (house)

Cycle 6 = I like SKIP the red house
```

Figure 8.4: Example Hypothesis Generation with 'following SKIPs' Problem

The third modification to the hypothesis generation process deals with the problem of combined repairs such as "the house | the red house | I mean red car". This is not the same as an embedded repair as one repair is not inside the other. It is, however, difficult as both repairs share some words. In general, if part of the *correction* of the first SKIP is equal to the front of the *reparandum* of the second SKIP, then it is possible that there are two SKIPs very close together. In this example "the red house" replaces "the house", but the "red house" is then replaced by the phrase "red car". Using conditional SKIPs, in the normal manner, the first repair (the house | the red house) could only be used if followed by the words "the red house". The second repair could not, therefore, be solved at the same time as the first repair as it can't be attached to the end of the phrase "SKIP

the red house". This problem can be seen in figure 8.4 where at cycle 4, when the second SKIP is required, the algorithm only allows the word "red" to be added to the hypothesis. It is not possible to use both SKIPs within the same hypothesis.

To overcome this a SKIP is allowed to follow another SKIP as long as the front of the *reparandum* of the second SKIP matches the end of the *correction* of the first SKIP. In the above example the second SKIP (red house I mean | red car) includes the end of the *correction* of the first SKIP (red house) at the front of its *reparandum*. Therefore, the second SKIP can replace "red house" within the *correction* of the first SKIP. This can be seen in figure 8.5 where on cycle 4 the *reparandum* of the second SKIP (red house) covers the remaining *correction* of the first SKIP. It is possible for the second SKIP to follow on from the first SKIP. This allows the second SKIP to replace the stored list and form its own stored list. Both possibilities will be automatically followed together with the normal word sequences.

```
Sentence = I like the house |  the red house I mean |  red car
Repair 1 =       (SKIP    ) | (the red house)
Repair 2 =                          (SKIP          ) | (red car)

Cycle 1 = I

Cycle 2 = I like

Cycle 3 = I like SKIP (the red house)

Cycle 4 = I like SKIP the (red house)

——— Overlap ———

Cycle 5 = I like SKIP the red (house)
          I like SKIP the SKIP (red car)

Cycle 6 = I like SKIP the red house
          I like SKIP the SKIP red (car)

etc.
```

Figure 8.5: Example Hypothesis Generation with 'following SKIPs' Processing

The 'following SKIPs' problem is important as it is often the case that one repair

closely follows another. If only conditional SKIPs were allowed many repairs would not be corrected. This modification allows overlapping SKIPs to be corrected.

Using the extra knowledge on the SKIPs added to the word lattice (see table 8.5), within the hypothesis generation process, after cycle 4, the hypothesis list shown in table 8.8 will include the following hypotheses:

SKIP the first quest
SKIP the fresh quest
SKIP the first question
SKIP the fresh question
SKIP first question dance
SKIP first question tan
SKIP first question to
SKIP first question ton
SKIP fresh question dance
SKIP fresh question tan
SKIP fresh question to
SKIP fresh question ton

This shows that the algorithm presented in this chapter can help solve speech repairs. It can be seen that the phrase the speaker actually intended to say (the first question) will appear within the hypothesis list.

8.4 Chapter Summary

This solution has been implemented within the AURAID system giving the structure shown in figure 8.6 and shows some promising results. The fact that this solution deals with repairs at the word lattice level means that the solution is not specific to AURAID, but rather, could be used by any automatic speech recognition system that uses a word lattice structure. Only two minor modifications would be required to the lattice parsing algorithm used by the system.

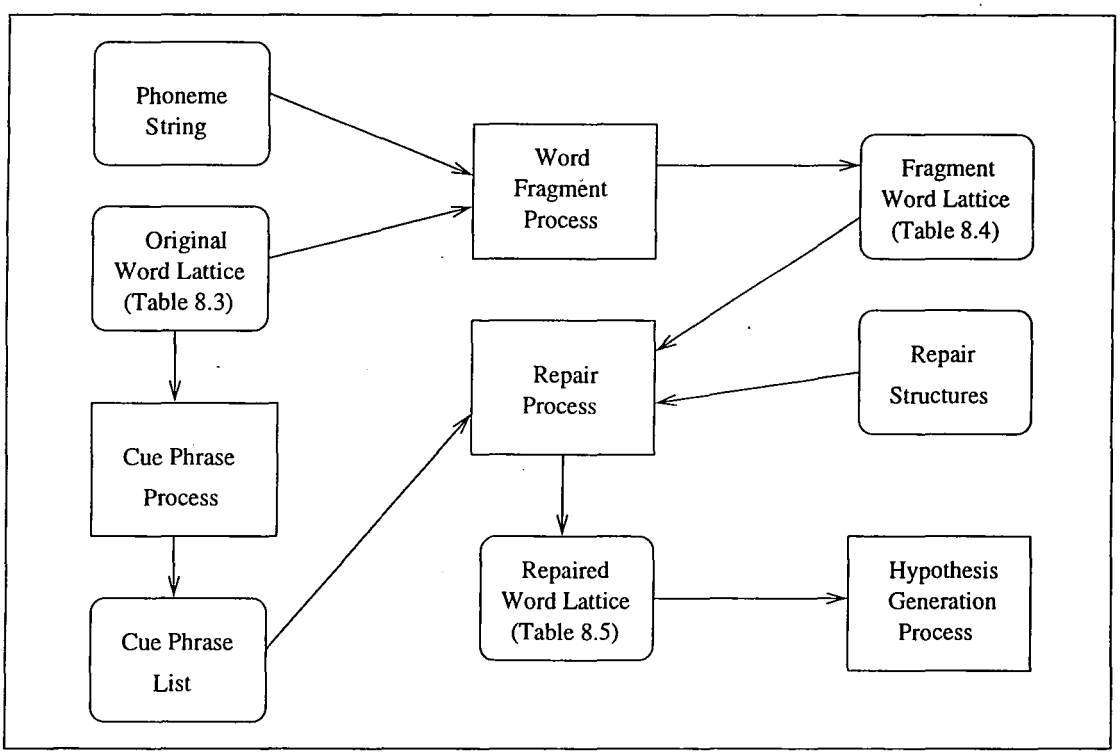


Figure 8.6: Block Structure of the Repair Process Within AURAID

Chapter 9

Results

This chapter discusses the results of four analyses carried out on the AURAID system. The system incorporates the approach to tackling speech repairs discussed in this thesis. The four analyses include both black box and white box approaches to allow various forms of evaluations to be performed on the system.

9.1 Introduction

As explained earlier (section 1.3), measuring the performance of a module that deals with speech repairs within a speech recognition process is not an easy task. For this reason four different sets of test data were produced and each used in a way that can help measure the performance of the repair module and the whole system. Using these different sets of data four different analyses were carried out:

- Specific Repair Analysis
- Repair Sentence Analysis
- Seen Passage Analysis
- Un-seen Passage Analysis

The first analysis (specific repair analysis) allows us to determine how each repair is dealt with and whether the repair process is giving the potential for the repair to be corrected.

The second analysis (repair sentence analysis) allows us to examine how each repair is dealt with at a sentence level. We can also determine the performance of the repair process as a whole by identifying false positives and the overall recognition rates of the system.

The third analysis (seen passage analysis) allows us to measure the performance of the whole system with respect to a large passage of data. The passage was part of the data which was analysed as a precursor to the production of the repair module. The performance measurements include recall and precision rates as well as the normal accuracy measurements of word error, words correct and word accuracy. Though not as accurate as the results of the first two analyses, it is these measurements that are used by other researchers and are, therefore, necessary for comparison purposes.

The fourth analysis (un-seen passage analysis) allows the measurement of the performance of the system on data that is, at present, un-seen and has not been used in any part of the analyses carried out with respect to producing the repair module. This gives recall and precision rates as well as the normal speech recognition measurements on the whole of the system.

9.2 The Dictionary

The analyses presented in this section used the extended dictionary of 2,275 words (see section 4.2.3 and table 4.4).

repair type	increase	no change	decrease	totals
1	17	-	-	17
2	3	1	1	5
3	4	2	2	8
totals	24	3	3	30

Table 9.1: Repair Type and System Performance: comparing the system without repair processing and the system with repair processing

9.3 Specific Repair Analysis

This first analysis investigates the effect of the repair process on the "exact" performance of the system, with respect to the required hypothesis. The measurements were those used in other analyses (see section 4.4). They are very accurate and show the true performance of the system.

9.3.1 The Data

The thirty repair passages used in the repair analysis (see chapter 5 and table 5.1) were processed using the modified and un-modified system and phoneme corruption of 15%. Therefore, two runs per passage were carried out:

- 15% phoneme corruption with no repair process.
- 15% phoneme corruption with repair process.

A comparison was made between those runs that used the repair process and those runs that did not use the repair process.

9.3.2 The Results

Table 9.1 compares the overall performance of the systems. It shows that twenty four out of the thirty passages resulted in an increased performance when repair

Pass	Without Repair Processing	With Repair Processing	Diff	+/-	Rank	Signed Rank
9	—	—	0	-	1	-2
20	—	—	0	-	2	-2
23	—	—	0	-	3	-2
1	—	1	+*	+	4	11.5
2	—	8	+*	+	5	11.5
3	—	2	+*	+	6	11.5
7	—	3	+*	+	7	11.5
8	—	3	+*	+	8	11.5
10	—	1	+*	+	9	11.5
11	—	1	+*	+	10	11.5
13	—	1	+*	+	11	11.5
14	—	13	+*	+	12	11.5
16	—	7	+*	+	13	11.5
18	—	2	+*	+	14	11.5
21	—	13	+*	+	15	11.5
22	—	1	+*	+	16	11.5
25	—	1	+*	+	17	11.5
27	—	1	+*	+	18	11.5
28	—	5	+*	+	19	11.5
15	4	1	3	+	20	20
4	5	1	4	+	21	21
17	12	19	-7	-	22	-22
29	54	1	53	+	23	23
26	74	12	62	+	24	24
5	156	13	143	+	25	25
19	166	3	163	+	26	26
12	153	488	-335	-	27	-27
24	596	29	567	+	28	28
30	1254	4	1250	+	29	29
6	193	—	-*	-	30	-30

Table 9.2: The Rank, Scores and Differences used in the Sign Test and Wilcoxon Signed Rank Test for the comparison of the system without repair processing and the system with repair processing

processing was added. By increased performance we mean that the required hypothesis was better placed within the hypothesis list after the processing of the passage.

Table 9.2 shows for each passage (Pass), the position of the required hypothesis when the two systems processed the repair sentences (a position of — shows that the required hypothesis was not in the hypothesis list). It also shows the difference (Diff) between the two positions[1][2], the direction of the difference (+/-), the rank of the difference (Rank), taking only the magnitude of the difference into account (i.e. ignoring the sign), and the signed rank score (Signed Rank) used in the Wilcoxon Signed Rank test.

The Sign Test[3]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with repair processing is better than the system without repair processing. A one-tail test is therefore appropriate.

Table 9.2 shows that the number of pairs that result in a positive outcome are 24 and the number of pairs that result in a negative outcome are 6. A difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with repair processing is better than the system without repair processing.

Using the Sign Test with a sample population of 30 the critical value for p=0.005 is 7. As the analysis has 6 negative outcomes, this shows that the system with repair processing is significantly better (6 is $\leq$ 7) than the system without repair

[1] A difference of -* shows that the exact difference can not be calculated but it is negative (i.e. the system with repair processing does not have the required hypothesis in the hypothesis list while the system without repair processing does).

[2] A difference of +* shows that the exact difference can not be calculated but it is positive (i.e. the system without repair processing does not have the required hypothesis in the hypothesis list while the system with repair processing does).

[3] See section 4.5, page 70, for a description of the Sign Test.

processing at the $p \ll 0.005$ level.

The Wilcoxon Signed Rank Test[4]

The NULL hypothesis will be that both systems are equal.

The alternate hypothesis will be that the system with repair processing is better than the system without repair processing. A one-tail test is therefore appropriate.

Table 9.2 shows the position (Rank) for each passage (Pass) when the difference (Diff) between the pairs is used to rank the sentences. As with the Sign Test a difference of zero is a negative outcome as it does not support the alternate hypothesis that the system with repair processing is better than the system without repair processing. An unknown difference, with the system with repair processing not having the required hypothesis in the hypothesis list (-*), is given the maximum rank as this is an undesirable result. An unknown difference, with the system without repair processing not having the required hypothesis in the hypothesis list (+*), is given a minimum difference of 1 as this is a desirable result. An unknown desirable result is given a minimum difference so as not to favour the unknown difference above its minimum potential. For those differences that are negative the results are kept negative while the others are positive.

The total of the positive ranks is 380 and the total of the negative ranks is 85.

Using the Wilcoxon Signed Rank Test with a sample population of 30 the critical value for $p=0.005$ is 109. As the analysis has a negative score of 85, this shows that the system with repair processing is significantly better (85 is $\leq$ 109) than the system without repair processing at the $p \ll 0.005$ level.

These tests demonstrates that repair processing is of benefit to the system when repairs are present.

[4]See section 4.5, page 70, for a description of the Wilcoxon Signed Rank Test.

Passage Analysis

As an illustration of how the repair processing is helping, the three passages described in detail within the repair analysis (see chapter 5) are explained in more detail.

Passage 1

The results of the processing for the first passage can be seen in table 9.3.

The required hypothesis was expanded beyond the *interruption point* of the repair, when the repair process was used. This did not occur when no repair processing was undertaken. It should also be noted that the required hypothesis was never top of the list. This shows that the required hypothesis would never have been selected as the most appropriate hypothesis by the current system. This raises two issues.

	No Repair Process					**Repair Process**				
	like a a typical economics book					like SKIP a typical economics book				
word	*top s*	*p*	*score*	*phrase*	*% dif*	*top s*	*p*	*score*	*skip*	*% dif*
	15% corruption									
like	126.0	2	126.0	0	0	126.0	2	126.0	0	0
SKIP/a	119.0	2	119.0	0	0	118.5	2	118.5	0	0
a	141.4	528	212.0	10	49	111.5	2	111.5	1	0
typical	-	-	-	-	-	95.5	2	95.5	1	0
economics						52.2	2	52.2	1	0
book						53.3	3	53.3	1	0

Table 9.3: Evaluation of the Repair Process: Passage 1

The first is the accuracy of the measurements being carried out. Because the measurements are very accurate what would normally be seen as a satisfactory result may seem a bit disappointing. The fact that the required hypothesis would not have been selected is not a problem. The thing to note is that the *skip* measurement (table 9.3), which shows the rank of the best hypothesis that uses the SKIP required to correct the repair, is one (i.e. the top hypothesis). Therefore,

	No Repair Process					Repair Process				
	describing the if you like the central part					describing SKIP the central part				
word	*top s*	*p*	*score*	*phrase*	*% dif*	*top s*	*p*	*score*	*skip*	*% dif*
	15% corruption									
describing	159.7	1	159.7	0	0	159.7	1	159.7	0	0
SKIP/the	145.7	1	145.7	0	0	145.7	2	150.7	4	3
if	124.7	468	272.0	0	118					
you	-	-	-	-	-					
the						124.7	22	142.6	22	14
central						103.7	19	119.4	19	15
part						89.7	3	98.4	3	9

Table 9.4: Evaluation of the Repair Process: Passage 2

the actual correction required for the repair was successfully carried out in the top hypothesis, even though it was not the required hypothesis.

The second issue is that some words have several tags and the system may not be able to choose between the tags, with the knowledge currently available to it. The actual output of the system was "like SKIP a typical economics book". However, the "like" was a verb (VERB) rather than a preposition (PREP). Therefore, the system selected the third hypothesis as the required hypothesis since this had the word like as a preposition (PREP). This shows that semantic and pragmatic knowledge would, probably, be helpful within the system. As the anti-grammar only punishes ill-formed constructs it is not able to favour preferable constructs. Semantic and pragmatic processing could, therefore, overcome this accuracy problem by re-scoring the hypothesis list giving a preference to semantically and pragmatically correct hypotheses.

Passage 2

The results of the processing for the second passage can be seen in table 9.4.

Of note here is that the addition of a *cue phrase* does not cause a problem. The repair process takes into account the possibility of a *cue phrase* directly after an *interruption point.*

	No Repair Process					Repair Process				
	the ques- the first question to answer					SKIP the first question to answer				
word	*top s*	*p*	*score*	*phrase*	*% dif*	*top s*	*p*	*score*	*skip*	*% dif*
	15% corruption									
SKIP/the ques-	168.0	1	168.0	1	0	159.2	10	159.2	10	0
the	-	-	-	-	-	145.2	1	145.2	1	0
first						117.2	10	117.2	10	0
question						84.7	1	84.7	1	0
to						70.7	1	70.7	1	0
answer						72.3	-	-	1	-

Table 9.5: Evaluation of the Repair Process: Passage 3

Once again the process goes beyond the *interruption point* when the repair module is used and the required hypothesis is near to the top. Without the repair module this did not occur and the required hypothesis never existed within the hypothesis list.

Passage 3

The results of the processing for the third passage can be seen in table 9.5.

The problem posed by a word fragment can be clearly seen. Without a repair mechanism the required hypothesis can't be followed once the location of the word fragment is reached. When repair processing is added and the word fragment identification system is used. The system successfully overcomes the repair. The repair is successfully negotiated during the run. The final outcome of the process is not the required string, this is due to the fact that there was heavy corruption on the final word, which was never recognised. The table shows that even though the system has a problem with the final word, the SKIP required to overcome the repair is used in the top hypothesis.

9.3.3 Conclusion

The results of this analysis show that the repair process is increasing the likelihood of the required hypothesis being selected, by creating the necessary entries and allowing the required hypothesis to exist within the hypothesis list. All three illustrative examples show how the repair process has allowed the repair to be overcome by including a SKIP into the word lattice.

9.4 Repair Sentence Analysis

The second analysis processes a number of sentences from the Durham data (chapter 7) which were used in producing the repair module. The exact performance of the system using the repair process was compared to the system without the repair process. The location of the required hypothesis was noted as in the previous analysis. In addition a more black box measurement was taken. This allowed the more traditional recall and precision measures to be used to measure the accuracy of the repair process on the final outcome of the recognition process. Further to this the more conventional measurements of word error, words correct and word accuracy rates were calculated to measure the overall performance of the system.

9.4.1 The Data

The first fifty repairs from the Durham data were used in this analysis. This resulted in forty four sentences being processed by the system. The fifty repairs were made up of twenty one type 1 repairs, thirteen type 2 repairs and eight type 3 repairs. Two of the type 1 repairs contained *cue phrases* and eight of the fifty repairs contained matching word fragments. The word fragments were contained in four type 1 repairs, two type 2 repairs and two type 3 repairs.

Three levels of phoneme corruption were used (0%, 15% and 25%) to see the effect of various levels of corruption. Six runs per sentence were carried out:

- 0% phoneme corruption with no repair process.
- 0% phoneme corruption with repair process.
- 15% phoneme corruption with no repair process.
- 15% phoneme corruption with repair process.
- 25% phoneme corruption with no repair process.
- 25% phoneme corruption with repair process.

9.4.2 The Results

Skip Existence

The first stage of the analysis involves determining whether the necessary SKIPs exist within the word lattice. As the SKIPs are conditional, to identify the existence of the required SKIP we need to find a SKIP which spans the un-required phonemes/words and is preceeded by the required words. Table 9.6 shows that the SKIP creation process is successful, in that it produces the required SKIP in the majority of cases and only fails when the corruption is so high that the phoneme string is far from the original transcription.

corruption	repairs	skips	percentage
0%	50	50	100%
15%	50	50	100%
25%	50	48	96%

Table 9.6: Performance of the Skip Creation Process

Required Hypothesis Measures

A comparison between the performance of the two systems shows a definite increase when the repair process is used. Table 9.7 shows the performance changes in the

Repair Type	0% corruption			15% corruption			25% corruption		
	inc.	-	*dec.*	*inc.*	-	*dec.*	*inc.*	-	*dec.*
1	21	8	0	20	9	0	19	7	3
2	4	3	1	4	3	1	3	3	2
3	7	3	3	6	5	2	5	6	2
Total	32	14	4	30	17	3	27	16	7
(%)	64%	28%	8%	60%	34%	6%	54%	32%	14%
Sent.	26	14	4	24	17	3	23	15	6
(%)	59%	32%	9%	54%	39%	7%	52%	34%	14%

Table 9.7: Accurate Performance on Repair Sentences

systems when the repair process is used. The figures have been split into repair types and show that no specific repair causes the most problems. The *Total* figures are for the repair types, there are fifty repairs, while the *Sent.* figures are for the actual sentences and there are forty four of these. In sentences containing more than one repair each repair type within the sentence is classed as an increase if the sentence showed an increased performance.

Table 9.7 shows that for 0% corruption 64% of the sentences showed an increased performance when repair processing is used. Similar increases are also seen for 15% and 25% corruptions. Actual decreases in performance were only found in 8%, 6% and 14% of the sentences.

The figures in table 9.7 compare the different systems. However, no difference in the performance could mean, either, both systems produced the required result or neither managed to produce the required result. In addition if the original system recognised the sentence containing the repair and the repair process corrected the repair. This was not a performance increase, but is a highly desirable result. Therefore, an additional measurement could be a desirable outcome measure: desirable, in the sense that the repair is overcome, even though the correct outcome may not be available. These figures are given in table 9.8 and show that the repair process produces the desired result in 77% to 89% of the sentences.

	desirable	undesirable
00	39 (89%)	5 (11%)
15	38 (86%)	6 (14%)
25	34 (77%)	10 (23%)
Avg.	37 (84%)	7 (16%)

Table 9.8: Repair Sentence : Desirable Results

Repair Measurements

The results so far have been based on the final result of the system. This relies on the rest of the system performing well and does not just measure the performance of the repair process. Of particular interest to this research is the number of times the repair process actually corrects a repair. This is determined by using the repair process performance measurements of recall and precision rates. Four separate measurements will be taken:

- Recognition Recall
- Recognition Precision
- Correction Recall
- Correction Precision

See section 1.3, page 6, for details of these measurements.

Table 9.9 shows the recall and precision rates with respect to the fifty repairs processed in the forty four test sentences. As can be seen between 70% and 90% of the actual repairs were corrected and between 68% and 90% resulted in the correct result. The precision rates are also very high as the false positive rate (3, 2 and 4 respectively) is very low. This is a key feature since false positives are a major problem within speech repair processing.

	0% corruption	15% corruption	25% corruption	Average
Repairs made	48	45	39	44
Actual repairs	45	43	35	41
Valid Corrections	45	43	34	40.7
False Positives	3	2	4	3
Recognition recall	90%	86%	70%	82%
Recognition precision	94%	96%	90%	93%
Correction recall	90%	86%	68%	81%
Correction precision	94%	96%	87%	92%

Table 9.9: Repair Sentence : Recall & Precision Rates

Classical measures

In order to determine the overall performance of a speech recognition system it is necessary to examine the word error, words correct and word accuracy rates. These measures are used extensively in speech recognition research.

Table 9.10 shows that the inclusion of the repair process increases the performance of the system[5]. The inclusion of SKIPs into the word lattice does not confuse the system, but allows it to select the most appropriate path through the lattice. This path could include both normal passages of text and repaired passages.

These figures show that the repair procedure is increasing the overall performance of the system (on average the accuracy is 4% better) and that there is little confusion taking place.

9.4.3 Conclusion

The results of this analysis show that the repair module is creating the necessary information to overcome the repairs, in the majority of the sentences. The high

[5]It must be noted that of the 50 repairs only 63 extra words are added to the sentences and during repair processing 50 SKIPs will be added to the sentences resulting in only 13 of the original 468 words in the sentences being deleted. This decreases the words in the sentences to 455 when the repair process is being used and this is taken into account in the WE, WC and WA measurements.

	0% corruption	15% corruption	25% corruption	Average
Repair Run				
Insertions	2	11	16	9.6
Deletions	21	16	12	16.3
Substitutions	32	46	54	44
Un-Repaired Run				
Insertions	4	7	19	10
Deletions	26	19	17	20.6
Substitutions	53	64	74	63.6
Results				
WE - repaired	12.1	16.0	18.0	15.4
actual	17.1	18.5	20.6	18.7
Difference	-5.0	-2.5	-2.6	-3.3
WC - repaired	88.3	86.3	85.5	86.7
actual	83.7	82.9	81.3	82.6
Difference	4.6	3.4	4.2	4.1
WA - repaired	87.9	84.0	82.0	84.6
actual	82.9	81.5	77.4	80.6
Difference	5.0	2.5	4.6	4.0

Table 9.10: Repair Sentence : Word Error, Words Correct & Word Accuracy Rates

recall rates show that this information is being used by the system to correct the repairs. The high precision rates and word accuracy figures show that the repair module is not confusing the hypothesis generation process.

9.5 Seen Passage Analysis

The third analysis used half of the original Durham data to investigate the effect of the repair process on a mixture of sentences with and without repairs. As the repair process was not given any assistance in identifying repairs or any specific information on the location of *cue phrases*, word fragments or edit signals, it is possible and likely that the repair process will identify possible repairs, within a lattice, for a non-repaired passage. Measurements are needed to identify the effect of the repair process on both repaired and correct sentences.

now the first thing to tell you is the book the recommended book for this course <.> is that <.> software engineering the third edition <.> don't get the first or the third however cheap it is it's awe- they're awful <.> it's this book here <.> it's eighteen or nineteen pounds but your all got plenty of money so you can all afford it <.> what i try to do on the course is that i don't exactly follow what's in that book <.> you should see this book as supplementary reading <.> i assume that you're reading the relevant sections and <.> occasionally i will point out the chapter you should read that i don't have time to cover <.> i should also point out that you don't have to know everything that's in this book <.> you have to know the sections in the book that i cover in the lectures and the sections that i point out each week <.> so there are bits in the book that i i i leave out altogether <.> and of course since your all very keen <.> erm here are some other books that you can go away and read as well <.> not don't buy any of these <.> couple of general software engineering books <.> in fact the the book by by prestman was the recommended book a couple of years back <.> the third one on the list the mythical man month common interesting contains interesting and funny stories about how software doesn't work <.> and why it falls over <.> and basically gives you some background information on why were here on this course <.> and the last book is software engineering economics <.> it's like a a typical economics books <.> it's a huge great thick book full of graphs and equations and and unusually for an economics book makes a lot of sense <.> but most of the stuff in this book is too far advanced for this course ...

Figure 9.1: Example Speech from the Seen Passage

9.5.1 The Data

Two sections of the Durham data (see figure 9.1 for an example of the speech), used in the corpus analyses presented in this thesis, were used in this analysis. The first quarter of the Durham data and the final quarter of the Durham data were added to form a single passage[6][7]. This passage was two hundred and twenty two sentences long and contained 2,544 words. Fifty six repairs were present within the passage. Of these fifty six repairs thirty four were type 1 repairs, nine were type 2 repairs and thirteen were type 3 repairs. Six of these fifty six repairs contained matching word fragments. As well as the fifty six repairs the passage also contained fourteen single filled pauses. These filled pauses were included as repairs for this analysis.

[6]Some pauses were removed from the middle of sentences.

[7]The first quarter and final quarter were used to ensure that speech from different times in the process of a lecture, were used in the analysis.

The same three levels of phoneme corruption, used in the "Repair Sentence Analysis" (see section 9.4) were used in this analysis, giving the same six runs.

	0% corruption	15% corruption	25% corruption	Average
Repair Type 1 (34)				
Repairs made	33	30	29	30.7
Valid Correction	33	30	27	30
Invalid Correction	0	0	2	0.7
Repair Type 2 (9)				
Repairs made	9	7	3	6.3
Valid Correction	7	5	3	5
Invalid Correction	2	2	0	1.3
Repair Type 3 (13)				
Repairs made	9	7	4	6.7
Valid Correction	9	7	4	6.7
Invalid Correction	0	0	0	0
Filled Pauses (14)				
Repairs made	11	10	9	10
Valid Correction	10	9	7	8.7
Invalid Correction	1	1	2	1.3
Totals				
Repairs made	90	85	75	83.3
Actual repairs	62	54	45	53.7
Valid Corrections	59	51	41	50.3
Invalid Corrections	3	3	4	3.3
False Positives	28	31	30	29.7
Recognition recall	89%	77%	64%	77%
Recognition precision	69%	64%	60%	64%
Correction recall	84%	73%	59%	72%
Correction precision	66%	60%	55%	60%

Table 9.11: Seen Passage : Recall & Precision Rates

9.5.2 The Results

The first set of results deal with the recall and precision of the repair process. Table 9.11 gives the figures for these calculations. This shows that an increase in the number of non-repaired sentences has resulted in a decrease in the accuracy of the process (when compared to the results in table 9.9). This is as would

be expected. The system still manages to deal with the majority of the speech repairs and though the false positive rate is significant it is not too large to warrant the process inadequate in dealing with repairs. It is expected that knowledge on semantics and pragmatics would help overcome some of these false positives, but this is beyond the scope of the work presented in this thesis.

A further point of interest is the decrease in performance as the corruption rate increases. This is again expected as when the corruption rate is high the system as a whole is unlikely to achieve high results. This decrease can be seen from the figures in table 9.12. The overall accuracy of the system decreases from 88/89% to 70%. This is a large drop and can account for the decreased performance of the repair process. If the system does not identify a word as appearing within the phoneme transcription then it will not appear in the word lattice and the repair process will not be able to use this word and hence it will fail. This is not an unexpected side effect and does not decrease the value of the repair process. However, it does show that a high phoneme recognition rate is required for the recognition system to be useful.

A feature which requires further study is the effect of false positives on the overall performance of the system. The small proportion of words contained within the repairs (3.7%[8]) means that any increase in accuracy of the system will be limited (with the introduction of the repair process) and the main figures of interest are the repair recall and precision values.

Table 9.12 shows the word error (WE), words correct (WC) and word accuracy (WA) figures for the different runs. As can be seen the overall performance of the system fluctuates slightly, but does not decrease dramatically. This shows that the repair process is not detrimental to the system and in fact increases the accuracy slightly (with an average accuracy increase of 0.4%).

The repair types that were corrected correctly were spread over the different

[8]It should be noted that the 70 repairs/filled pauses added only 94 extra words to the passage and during repair processing 70 SKIPs will be added to the passage resulting in only 24 (0.9%) of the original 2,544 words in the passage being deleted. This decreases the words in the passage to 2,520 when the repair process is applied.

	0% corruption	15% corruption	25% corruption	Average
Repair Run				
Insertions	19	81	131	77
Deletions	80	82	89	83.7
Substitutions	176	377	530	361
Un-Repaired Run				
Insertions	8	80	131	73
Deletions	100	95	84	93
Substitutions	195	379	538	370.7
Results				
WE - repaired	10.9	21.4	29.8	20.7
actual	11.9	21.8	29.6	21.1
Difference	-1.0	-0.4	0.2	-0.4
WC - repaired	89.8	81.8	75.4	82.4
actual	88.4	81.4	75.6	81.8
Difference	1.4	0.4	0.2	0.6
WA - repaired	89.0	78.6	70.2	79.3
actual	88.0	78.2	70.4	78.9
Difference	1.0	0.4	-0.2	0.4

Table 9.12: Seen Passage : Word Error, Words Correct & Word Accuracy Rates

repair types (including filled pauses, see table 9.11).

The decrease in accuracy for the more complex repairs is likely to be due to the larger number of words and phonemes contained within the repairs. This increases the likelihood of a corruption occuring within the repair. Therefore, the words of the repair will not be recognised and the repair can not take place.

9.5.3 Conclusion

The results of this analysis show that the repair process is increasing the likelihood of the correct passage being selected. Repairs are being identified and corrected and the false positives are being kept to an acceptable level. Taking into account the level of specific knowledge given to the system (i.e. no word fragments identified, no *cue phrases* identified and no filled pauses identified) these are very encouraging

results and show that the repair process can help overcome speech repairs.

9.6 Un-seen Passage Analysis

The fourth and final analysis examines the effect of the repair process on un-seen data. Data that was not used in the analyses carried out for this research was used as test data for the system. This was investigated to determine whether the solution and findings are specific to the data analysed.

okay <.> well <.> i'd certainly like to welcome you all this evening to durham <.> i know you've travelled quite a way from teesside and newcastle <.> it might be good to explain about the title of our talk this evening <.> and really i've split the talk into four parts <.> i'll give you a a bit of background information about erm our work within the school of engineering and computer science at durham <.> and i'll go on to erm give you some brief discussion of why i think processes are important <.> erm both in their own right and as basically a reaction <.> software processes anyway <.> as a reaction to the erm business process demands that are currently placed on i-t departments <.> erm i shall give you some of an introduction on the software process modelling research work we've done at durham <.> and erm go on to discuss how the approach can be more generally applied to process improvement <.> and erm well how it's changed management generally speaking on on business processes <.> within the school of engineering and computer science <.> in the computer science group there are two research groups <.> i work in the centre for software maintenance <.> which was established in nineteen eighty seven <.> erm over that period since since eighty seven we've had some key achievements <.> basically <.> we've established here at durham an international workshop on an annual basis <.> which now gets one hundred and sixty people attending it <.> not just researchers but practitioners interested in in software maintenance <.> this is a really hot topic in software engineering <.> and we we have also established a journal of software maintenance ...

Figure 9.2: Example Speech from the Un-Seen Passage

9.6.1 The Data

An un-seen passage (see figure 9.2 for an example of the speech from the un-seen passage) taken from the Durham corpus, was used in this evaluation[9]. This passage

[9]Some pauses were removed from the middle of sentences.

was the first half of a talk given at a British Computer Society meeting held at the University of Durham. It introduces and explains a project on a software engineering topic.

	0% corruption	15% corruption	25% corruption	Average
Repair Type 1 (115)				
Repairs made	94	90	83	89
Valid Correction	87	89	83	86.3
Invalid Correction	7	1	0	2.7
Repair Type 2 (12)				
Repairs made	10	5	7	7.3
Valid Correction	9	5	6	6.7
Invalid Correction	1	0	1	0.7
Repair Type 3 (8)				
Repairs made	5	5	4	4.7
Valid Correction	4	4	3	3.7
Invalid Correction	1	1	1	1
Filled Pauses (60)				
Repairs made	50	46	44	46.7
Valid Correction	44	42	40	42
Invalid Correction	6	4	4	4.7
Totals				
Repairs made	226	187	178	197
Actual repairs	159	146	138	147.7
Valid Corrections	144	140	132	138.7
Invalid Correction	15	6	6	9
False Positives	67	41	40	49.3
Recognition recall	82%	75%	71%	76%
Recognition precision	70%	78%	78%	75%
Correction recall	73%	72%	68%	71%
Correction precision	64%	75%	74%	71%

Table 9.13: Un-Seen Passage : Recall & Precision Rates

The passage was approximately thirty minutes long and was made up of three hundred and sixty three sentences containing 3,470 words. One hundred and thirty five repairs (type 1, 2 and 3 disfluencies) were present within the passage. Of these, one hundred and fifteen were type 1 repairs, twelve were type 2 repairs and eight were type 3 repairs. Nine of these repairs contained word fragments (all type 1

repairs and all with matching words) and fifteen contained *cue phrases.* Of the repairs containing *cue phrases* six were type 1, four were type 2 and five were type 3 repairs. As well as the repairs the passage also contained sixty single filled pauses. These filled pauses were included as repairs for this analysis.

Again the same three levels of corruption were used (as in the two previous analyses) giving the same six runs.

9.6.2 The Results

Table 9.13 gives the recall and precision rates for this test and table 9.14 shows the word error (WE), words correct (WC) and word accuracy (WA) measures.

The figures show that the overall accuracy of the system is not as good as with the seen data, but this is due to the speech and not the repair process. However, the system performs on average 2.8% better on word accuracy measures when the repair process is used[10]. This shows that the repair process aids with the overall performance of the system and that the false positives are being kept to a minimum.

The recall and precision rates (table 9.13) are slightly lower than for the seen data, but this is expected. There are three repairs in the un-seen data that are not taken into account within the repair process. Even so the system still achieves averages of 76% recognition recall and 75% correction recall. Again, the performance of the system decreases as the corruption rate increases. This performance decrease is less dramatic than with the seen data. Though the system performs better on type 1 repairs than on others all repairs are corrected to an acceptable level.

[10]It should be noted that the 195 repairs/filled pauses added only 214 extra words to the passage and during repair processing 195 SKIPs will be added to the passage resulting in only 19 (0.6%) of the original 3,470 words in the passage being deleted. This decreases the words in the passage to 3,451 when the repair process is applied.

	0% corruption	15% corruption	25% corruption	Average
Repair Run				
Insertions	54	142	218	138
Deletions	214	220	212	215.3
Substitutions	536	768	929	744.3
Un-Repaired Run				
Insertions	46	129	211	128.7
Deletions	232	254	246	244
Substitutions	580	873	1028	827
Results				
WE - repaired	23.3	32.7	39.4	31.8
actual	24.7	36.2	42.8	34.6
Difference	-1.4	-3.5	-3.4	-2.8
WC - repaired	78.3	71.4	66.9	72.2
actual	76.6	67.5	63.3	69.1
Difference	1.7	3.9	3.6	3.1
WA - repaired	76.7	67.3	60.6	68.2
actual	75.3	63.8	57.2	65.4
Difference	1.4	3.5	3.4	2.8

Table 9.14: Un-Seen Passage : Word Error, Words Correct & Word Accuracy Rates

9.6.3 Conclusion

The results of this analysis, again, show that the repair process is increasing the likelihood of the correct passage being identified. Repairs are being identified and corrected and the false positives are being kept to an acceptable level. It is encouraging to see that the repair process can work on un-seen data.

9.7 Chapter Summary

These four analyses cover various testing strategies which include both white box and black box testing. The "Specific Repair Analysis" shows that the required string (with the repair corrected) is more likely to be selected when the repair process is applied. This is the main measure of success as other knowledge, on things such as semantics and pragmatics, should aid in the selection of the required

string, once the repair has been corrected. These results were calculated using very accurate measurements including the word tags and the words themselves.

The "Repair Sentence Analysis" demonstrates that the repair process produces the required knowledge within the word lattice by creating between 96% to 100% of the required SKIPs. This test also shows that the recall and precision rates can be high.

The "Seen Passage Analysis" shows that although the system does identify possible repairs in correct speech, the frequency of these false positives is low. The increase in the word accuracy and words correct values between the system not using the repair process and the system using the repair process shows that the confusion is being kept to a minimum.

The "Un-Seen Passage Analysis" demonstrates that the system can cope with data that is unknown to the system, as far as repairs and repair structures are concerned. This again indicates that the repair process is not confusing the system and that the system can cope with the information produced by the repair process.

These four analyses show that the repair process developed using the grammatical knowledge from the corpus analyses (see chapter 7) can be used to successfully overcome repairs within spontaneous speech, without decreasing the overall performance of the system.

Chapter 10

Conclusions

This chapter summarises the project and identifies the extent to which the project meets its original goals. Those sections of the project that have resulted in the greatest impact on the field of Automatic Speech Recognition are identified and some future directions of research are discussed.

10.1 Project Summary

The aim of this project was to:

> ...incorporate into the sub-section of an automatic speech recognition system, being developed for the general problem of recognising spontaneous speech within a lecturing environment, a module to deal with speech repairs, using knowledge readily available to the system.

Identifying the criteria for the success of the project was a difficult task. The success of the project can not be measured simply on the final performance of the system, but must also consider the internal performance of the system with the location of the required hypothesis. Three levels of success have, therefore, been identified. The system must:

- Allow the possibility of the speaker's mistake to be corrected by the system.

- Increase the likelihood of the required output being selected.
- Identify and correct speech repairs.

The four different sets of test data were designed to test the system at all three levels and to test the system in both white box and black box modes. The results of the tests:

- Show that the repair process is adding SKIPs to the word lattice, allowing the system to choose between the original passage and a repaired passage. The system, therefore, has the potential to correct the speaker's mistakes.
- Show that the required hypothesis is higher in the hypothesis list (a higher position incudes a hypothesis that exists compared to a hypothesis that does not exist) when repair processing is used, compared to no repair processing, and is, therefore, more likely to be selected.
- Show that the SKIPs are being selected and used to the benefit of the overall performance of the system.

From these goals and results it can be seen that the project was successful in producing a repair module, which, when added to the AURAID system increased the performance of the system, when dealing with spontaneous speech, in both black box and white box measurements.

10.2 Impact on Automatic Speech Recognition

The project has developed an automatic process for the identification and correction of speech repairs within an automatic speech recognition system. Current research into repair processing has concentrated, mainly, on processing a string of text and identifying speech repairs or disfluencies within this string. Unfortunately we do not have speech recognition systems that could reproduce, to the required

accuracy, the exact words spoken by a human in a natural environment. Automatic speech recognition systems are becoming very accurate for read speech in a limited domain. However, they can not yet process spontaneous speech and all the problems spontaneous speech poses, e.g. disfluencies and repairs. This project has developed a method for identifying and correcting repairs before an automatic speech recognition system selects its final outcome, therefore, solving one of the problems posed by spontaneous speech to speech recognition systems.

This theory and solution have made the following advances:

- A process for identifying possible word fragments (those only matching repeated words) at a sub-word level has been developed and used successfully in the repair process. This process does not deal with all forms of word fragments, only those that match a repeated word (most of the word fragments found in our data were of this form). Word fragments are a major problem to speech recognition systems and can only be dealt with at a sub-word level. Moving to the phoneme transcription and adding "wild cards" to the word lattice, for later use in repair processing, results in a technique for tackling word fragments.

- A speech repair grammar has been identified and used successfully in the repair process. The repair grammar identified in this research is not likely to be a complete set of structures that make up all of the possible repairs to be found in speech. However, the process does show that a repair grammar can be used successfully by a repair process within a speech recognition system. The use of a full repair grammar could be incorporated into any speech recognition system.

- Sub-hypothesis level processing has proved successful. By incorporating the repair process into the system at a sub-hypothesis level, it shows that the theory can be incorporated into any automatic speech recognition system that uses a word lattice construct. Repair processing need not only be tackled at the word string level. The changes required to the AURAID system were minimal and the potential confusion of the inclusion of SKIPs into the

word lattice is also minimal. It demonstrates that the theory can be incorporated into a system with the minimal amount of effort and that word lattice processing seems to be a beneficial way of dealing with speech repairs.

10.3 Further Work

There are three main areas in which the research presented in this thesis could progress.

Extend the Data

The completion of the Durham corpus would allow a more complete grammar of repairs to be identified. The completion of the Durham corpus would allow the repair structures to be expanded so that a larger set could be produced and included into the system. Despite this we believe that the repair structures found, so far, would make up the majority of any repair grammar.

At present the AURAID system runs on a word class tag set of nine different classes. If this was expanded, the details of the repair grammar could be fine tuned so that less confusion would take place between correct passages and repair passages. This is not an essential addition but is something that would increase the accuracy of the repair process and eliminate a number of the false positives.

Semantic Knowledge

The addition of semantic information into the AURAID system would be beneficial in many ways. The required hypotheses should move closer (presuming the speaker is making sense) to the top of the hypothesis list, making them more likely to be selected. This should also be true for those hypotheses that have been correctly corrected by the repair process. Without repair processing a sentence containing a repair is not likely to 'make sense' and would, therefore, be penalised by a semantic process. Once the repair is corrected, the sentence should 'make sense' and the semantic process should not penalise this sentence.

It may also be possible to use semantic information to deal with type 4 and

type 5 disfluencies. Type 4 disfluencies come in various forms, but all require some form of link between the *reparandum* and *correction* before the true meaning can be ascertained. It is likely that semantic processing is the only way of dealing with this problem. Similarly type 5 disfluencies are meaning problems and can only be overcome if a meaning for the sentence can be formed. An investigation into this could complete the coverage of the speech disfluencies found in spontaneous speech.

Prosodic Knowledge

Prosodic or acoustic information can be helpful to the speech recognition process in a number of different ways. One of the benefits of using prosodic knowledge may be in the identification and correction of speech repairs. Prosodic knowledge has been used in the repair process by a number of different researchers and it is believed that prosodic information plays an important role in speech repairs. I agree with this theory and believe that the introduction of prosodics into the repair process will bring certain benefits. Prosodic information could be used to distinguish between a repair location and a false positive, thus eliminating many SKIPs and increasing the accuracy of the repair process. It is believed that the acoustics change when a word fragment is present and this information could be used to identify, more accurately, matching word fragments. If the acoustic change is significant, prosodic information may be used to identify the location of non-matching word fragments.

References

[Agnas *et al.*, 1994] M. Agnas, H. Alshawi, I. Bretan, D. Carter, K. Ceder, M. Collins, R. Crouch, V. Digalakis, B. Ekholm, B. Gamback, J. Kaja, J. Karlgren, B. Lyberg, P. Price, S. Pulman, M. Rayner, C. Samuelsson, and T. Svensson, "Spoken Language Translator: First Year Report", Technical Report 94-03, SRI International, January 1994.

[Allen *et al.*, 1994] J. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. G. Martin, B. W. Miller, M. Poesio, and D. R. Traum, "The TRAINS project: A case study in building a conversational planning agent", Technical Report 94-3, The University of Rochester, September 1994.

[Anderson *et al.*, 1991] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thomptson, and R. Weinert, "The HCRC Map Task Corpus", *Language and Speech*, 34(4):351–366, 1991.

[Baker, 1987] J. M. Baker, "State-Of-The-Art Speech Recognition U.S. Research and Business Update", in *Proceedings of the Eurospeech European Conference on Speech Communication and Technology*, volume 1, pages 440–447, 1987.

[Beach, 1991] C. M. Beach, "The Interpretation of Prosodic Patterns at Points of Syntactic Structure Ambiguity: Evidence for Cue Trading Relations", *Journal of Memory and Language*, 30:644–663, 1991.

[Bear *et al.*, 1992] J. Bear, J. Dowding, and E. Shriberg, "Integrating Multiple Knowledge Sources for Detection and Correction of Repairs in Human-Computer

Dialog", in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 56–63, June 1992, Delaware, USA.

[Bear *et al.*, 1993] J. Bear, J. Dowding, E. Shriberg, and P. Price, "A System for Labeling Self-Repairs in Speech", Technical Report 522, SRI International, February 1993.

[Bernstein and Danielson, 1992] J. Bernstein and D. Danielson, "Spontaneous Speech Collection for the CSR Corpus", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 373–378, Morgan Kaufmann, February 1992.

[Biber, 1987] D. Biber, *Textual Relations in Speech and Writing*, Cambridge University Press, 1987.

[Blaauw, 1992] E. Blaauw, "Phonetic Differences Between Read and Spontaneous Speech", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 751–754, October 1992, Alberta, Canada.

[Blackmer and Mitton, 1991] E. R. Blackmer and J. L. Mitton, "Theories of Monitoring and the Timing of Repairs in Spontaneous Speech", *Cognition*, 39:173–194, 1991.

[Broen and Siegel, 1972] P. A. Broen and G. M. Siegel, "Variations in normal speech disfluencies", *Language and Speech*, 15(4):219–231, 1972.

[Browning *et al.*, 1990] S. R. Browning, R. K. Moore, K. M. Ponting, and M. J. Russell, "A Phonetically Motivated Analysis of the Performance of the ARM Continuous Speech Recognition System", in *Proceedings of the Institute of Acoustics Speech and Hearing Conference*, November 1990, Windermere.

[Browning, 1993a] S. Browning, "Collecting a corpus of unscripted speech using a Wizard of Oz technique", Technical Report 4675, DRA Malvern, 1993.

[Browning, 1993b] S. Browning, "Collecting more spoken enquiries to an information system", Technical Report 4757, DRA Malvern, 1993.

[Butzberger *et al.*, 1992] J. Butzberger, H. Murveit, E. Shriberg, and P. Price, "Spontaneous Speech Effects In Large Vocabulary Speech Recognition Applications", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 339–343, 1992.

[Carletta *et al.*, 1993] J. Carletta, R. Caley, and S. Isard, "A Collection of Self-Repairs from the Map Task Corpus", Technical Report 47, HCRC, University of Edinburgh, November 1993.

[Carter and Rayner, 1994] D. Carter and M. Rayner, "The Speech-Language Interface in the Spoken Language Translator", in *Proceeding of Twente Workshop on Language Technology (TWENTE-8)*, December 1994, Holland.

[Chafe and Tannen, 1987] W. Chafe and D. Tannen, "The Relation Between Written and Spoken English", *Annual Review of Anthropology*, 16:383–407, 1987.

[Collingham and Garigliano, 1993] R. J. Collingham and R. Garigliano, "Using Anti-Grammar and Semantic Categories for the Recognition of Spontaneous Speech", in *Proceedings of the Eurospeech European Conference on Speech Communication and Technology*, pages 1951–1954, September 1993, Berlin.

[Collingham, 1994] R. J. Collingham, *An Automatic Speech Recognition System for use by Deaf Students in Lectures*, PhD thesis, University of Durham, 1994.

[Cutler, 1983] A. Cutler, "Speaker's Conceptions of the Functions of Prosody", in A. Cutler and D. R. Ladd, editors, *Prosody: Measures and Measurements*, Springer, Heidelberg, 1983.

[Daly and Zue, 1992] N. A. Daly and V. W. Zue, "Statistical and Linguistic Analyses of f0 in Read and Spontaneous Speech", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 763–766, October 1992, Alberta, Canada.

[Dowding *et al.*, 1993a] J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "GEMINI: A Natural Language System For Spoken-

Language Understanding", in *Proceedings of the ARPA Human Language Technology Workshop*, pages 43–48, March 1993, Priceton, NJ.

[Dowding *et al.*, 1993b] J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "GEMINI: A Natural Language System For Spoken-Language Understanding", in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 54–61, 1993.

[Edwards, 1993] J. A. Edwards, "Survey of Electronic Corpora and Related Resources for Language Researchers", in J. A. Edwards and M. D. Lampert, editors, *Talking Data: Transcription and coding in discourse research*, pages 263–306, Lawrence Erlbaum Associates, 1993.

[Fink and Biermann, 1986] P. K. Fink and A. W. Biermann, "The Correction of Ill-Formed Input using History-Based Expectation with Applications to Speech Understanding", *Computational Linguistics*, 12(1):13–36, 1986.

[Fox Tree, 1995] J. E. Fox Tree, "The Effects of False Starts and Repetitions on the Processing of Subsequent Words in Spontaneous Speech", *Journal of Memory and Language*, 34:709–738, 1995.

[Francis and Kucera, 1979] W. N. Francis and H. Kucera, *Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for Use with Digital Computers*, 1979.

[Garigliano *et al.*, 1993] R. Garigliano, K. Johnson, and R. J. Collingham, "A Data-Supported Case for a Spontaneous Speech Grammar", in *Proceedings of the Eurospeech European Conference on Speech Communication and Technology*, pages 969–972, ESCA, September 1993, Berlin.

[Gibson *et al.*, 1966] J. W. Gibson, C. R. Gruner, R. J. Kibler, and F. J. Kelly, "A Quantitative examination of differences and similarities in written and spoken messages", *Speech Monographs*, 33:444–451, 1966.

[Goffman and Hymes, 1981] E. Goffman and D. Hymes, *Forms of Talk*, chapter The lecture, pages 162–195, University of Pennsylvania Press, Philadelphia, 1981.

[Goldman-Eisler, 1961] F. Goldman-Eisler, "A Comparative study of two hesitation phenomina", *Language and Speech*, 4:18–26, 1961.

[Heeman and Allen, 1994a] P. A. Heeman and J. Allen, "Detecting and correcting speech repairs", in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 295–302, June 1994, Priceton, NJ.

[Heeman and Allen, 1994b] P. A. Heeman and J. Allen, "Tagging Speech Repairs", in *Proceedings of the ARPA Human Language Technology Workshop*, March 1994, USA.

[Heeman and Loken-Kim, 1995] P. A. Heeman and K. Loken-Kim, "Using Structural Information to Detect Speech Repairs", Technical Report TR IEICE SP95-91, ATR Interpreting Telecommunications Research Laboratories, December 1995.

[Heeman, 1994] P. A. Heeman, "Spoken Dialogue Understanding and Local Context", Technical Report 523, The University of Rochester, July 1994.

[Hindle, 1983] D. Hindle, "Deterministic Parsing of Syntactic Non-fluencies", in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 123–128, 1983.

[Hirschberg and Nakatani, 1993] J. Hirschberg and C. Nakatani, "A Speech-First Model For Repair Identification In Spoken Language Systems", in *Proceedings of the Eurospeech European Conference on Speech Communication and Technology*, volume 2, pages 1173–1176, September 1993, Berlin, Germany.

[Hirschman *et al.*, 1993] L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann, "Multi-Site Data Collection and Evaluation in Spoken Language Understand-

ing", in *Proceedings of the ARPA Human Language Technology Workshop*, pages 19–24, March 1993.

[Howell and Young, 1991] P. Howell and K. Young, "The Use of Prosody in Highlighting Alterations in Repairs from Unrestricted Speech", *The Quarterly Journal of Experimental Psychology*, 43A(3):733–758, 1991.

[Johansson and Hofland, 1987] S. Johansson and K. Hofland, "The Tagged LOB Corpus: Description and analysis", in W. Meijs, editor, *Corpus Linguistics and beyond*, pages 1–20, Rodopi B.V., Amsterdam, 1987.

[Johansson *et al.*, 1978] S. Johansson, G. Leech, and H. Goodluck, *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers*, 1978.

[Johnson *et al.*, 1994a] K. Johnson, R. J. Collingham, and R. Garigliano, "Data-Supported Case for the Extended Coverage of Repairs in the Recognition of Natural Speech", in *Proceedings of the Institute of Acoustics Autumn Conference : Speech and Hearing*, pages 31–38, November 1994, Windermere.

[Johnson *et al.*, 1994b] K. Johnson, R. Garigliano, and R. J. Collingham, "Data-Based Control of the Search Space Generated by Multiple Knowledge Bases for Speech Recognition", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 2147–2150, September 1994, Yokohama, Japan.

[Junqua and Haton, 1996] J. Junqua and J. Haton, *Robustness in Automatic Speech Recognition: fundamentals and applications*, Kluwer Academic Publishers, 1996.

[Klatt, 1977] D. H. Klatt, "Review of the ARPA Speech Understanding Project", *Journal of the Acoustical Society of America*, 62(6):1345–1366, December 1977.

[Koopmans-van Beinum, 1990] F. Koopmans-van Beinum, "Spectro-Temporal Reduction and Expansion in Spontaneous Speech and Read Text: The Role of Fo-

cus Words", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1990, Kobe, Japan.

[Labov, 1966] W. Labov, "On the grammaticality of everday speech", in *Linguistic Society of America Annual Meeting*, 1966.

[Lalljee and Cook, 1969] M. Lalljee and M. Cook, "An Experimental investigation of the function of Filled Pauses in Speech", *Language and Speech*, 12(1):24–29, 1969.

[Lamel *et al.*, 1986] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus", in *Proceedings of the DARPA Speech Recognition Workshop*, pages 100–109, February 1986.

[Lamel *et al.*, 1994] L. Lamel, F. Schiel, A. Fourcin, J. Mariani, and H. Tillmann, "The Translanguage English Database (TED)", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 1795–1798, September 1994, Yokohama, Japan.

[Lea, 1980] W. A. Lea, editor, *Trends in Speech Recognition*, Prentice Hall Signal Processing Series, Prentice Hall, Inc., 1980.

[Leech and Fligelstone, 1992] G. Leech and S. Fligelstone, "Computers and Corpus Analysis", in C. S. Butler, editor, *Computers and Written Texts*, pages 115–140, Blackwell, 1992.

[Levelt and Cutler, 1983] W. J. M. Levelt and A. Cutler, "Prosodic Marking in Speech Repair", *Journal of Semantics*, 2(2):205–217, 1983.

[Levelt, 1983] W. J. M. Levelt, "Monitoring and Self-Repair in Speech", *Cognition*, 14:41–104, 1983.

[Levelt, 1989] W. J. M. Levelt, *Speaking: From intention to articulation*, MA: M.I.T. Press, 1989.

[Lickley and Bard, 1992] R. J. Lickley and E. G. Bard, "Processing Disfluent Speech: Recognising Disfluency Before Lexical Access", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 935–938, 1992, Alberta, Canada.

[Lickley, 1994] R. J. Lickley, *Detecting Disfluency in Spontaneous Speech*, PhD thesis, University of Edinburgh, 1994.

[Liggett, 1984] S. Liggett, "The Relationship Between Speaking and Writing: An Annotated Bibliography", *College Composition and Communication*, 35(3):334–344, October 1984.

[Marcus *et al.*, 1993] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank", *Computational Linguistics*, 19(2):313–330, 1993.

[Marcus *et al.*, 1994] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank", in S. Armstrong, editor, *Using Large Corpora*, pages 273–290, MIT Press, 1994.

[Mitton, 1992] R. Mitton, *A Description of a Computer-Usable Dictionary File Based on the Oxford Advanced Learner's Dictionary of Current English*, June 1992.

[Morris, 1991] A. Morris, "A Wizard of Oz technique for capturing utterances of unscripted speech", Technical Report 181, DRA Malvern, 1991.

[Nakatani and Hirschberg, 1993a] C. Nakatani and J. Hirschberg, "A Speech-First Model For Repair Detection and Correction", in *Proceedings of the ARPA Human Language Technology Workshop*, pages 329–334, 1993, Plainsboro, NJ.

[Nakatani and Hirschberg, 1993b] C. Nakatani and J. Hirschberg, "A Speech-First Model For Repair Detection and Correction", in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 46–53, 1993, Columbus, OH.

[Nakatani and Hirschberg, 1994] C. Nakatani and J. Hirschberg, "A Corpus-Based Study of Repair Cues in Spontaneous Speech", *Journal of the Acoustical Society of America*, 95(3):1603–1616, March 1994.

[O'Donnell, 1974] R. C. O'Donnell, "Syntactic Differences Between Speech and Writing", *American Speech*, 49(1–2):102–110, 1974.

[O'Shaughnessy, 1992] D. O'Shaughnessy, "Analysis of False Starts in Spontaneous Speech", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 931–934, 1992, Alberta, Canada.

[O'Shaughnessy, 1993a] D. O'Shaughnessy, "Analysis and Automatic Recognition of False Starts in Spontaneous Speech", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 724–727, 1993, Minneapolis.

[O'Shaughnessy, 1993b] D. O'Shaughnessy, "Handling False Starts in the Recognition of Spontaneous Speech", *Canadian Acoustics*, 21(3):49–50, September 1993.

[O'Shaughnessy, 1994] D. O'Shaughnessy, "Correcting Complex False Starts in Spontaneous Speech", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 349–352, 1994.

[Oviatt, 1995] S. Oviatt, "Predicting Spoken Disfluencies During Human-Computer Interaction", *Computer Speech and Language*, 9:19–35, 1995.

[Paul and Baker, 1992] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-based CSR Corpus", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 899–902, October 1992, Alberta, Canada.

[Phillips *et al.*, 1992] M. Phillips, J. Glass, J. Polifroni, and V. Zue, "Collection and Analyses of WSJ-CSR Corpus at MIT", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 907–910, October 1992, Alberta, Canada.

[Price *et al.*, 1988] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 651–654, 1988, New York.

[Price *et al.*, 1991] P. Price, M. Ostendorf, S. Shattuck-Hufnagel, and C. Fong, "The Use of Prosody in Syntactic Disambiguation", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 372–377, February 1991.

[Rayner *et al.*, 1993] M. Rayner, H. Alshawi, I. Bretan, D. Carter, V. Digalakis, B. Gamback, J. Kaja, J. Karlgren, B. Lyberg, S. Pulman, P. Price, and C. Samuelsson, "A speech to Speech Translation system Built from Standard Components", in *Proceedings of the ARPA Human Language Technology Workshop*, pages 217–222, March 1993, Priceton, NJ.

[Rayner *et al.*, 1994] M. Rayner, D. Carter, V. Digalakis, and P. Price, "Combining Knowledge Sources to Reorder N-Best Speech Hypothesis Lists", in *Proceedings of the ARPA Human Language Technology Workshop*, March 1994, Priceton, NJ.

[Sagawa *et al.*, 1994] Y. Sagawa, M. Ito, N. Ohnishi, and N. Sugie, "A Model for Generating Self-Repairs", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 371–374, September 1994, Yokohama, Japan.

[Schegloff, 1979] E. A. Schegloff, "The Relevance of repair to Syntax for Conversation", in G. Talmy, editor, *Syntax and Semantics Volume 12: Discourse and Syntax*, Academic Press, New York, 1979.

[Shriberg *et al.*, 1992] E. Shriberg, J. Bear, and J. Dowding, "Automatic Detection and Correction of Repairs in Human-Computer Dialogue", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 419–424, February 1992, Harriman, N.Y.

[Shriberg, 1994] E. E. Shriberg, *Preliminaries to a theory of speech disfluencies*, PhD thesis, University of California at Berkeley, 1994.

[Siegel and Martin, 1967] G. M. Siegel and R. R. Martin, "Verbal Punishment of Disfluencies during Spontaneous Speech", *Language and Speech*, 10(4):244–251, 1967.

[Siegel, 1956] S. Siegel, *Nonparametric Statistics For The Behavioural Sciences*, McGraw-Hill Book Company, 1956.

[Silverman *et al.*, 1992a] K. E. A. Silverman, E. Blaauw, J. Spitz, and J. F. Pitrelli, "Towards Using Prosody in Speech Recognition/Understanding Systems: Differences Between Read and Spontaneous Speech", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 435–440, February 1992, Harriman, N.Y.

[Silverman *et al.*, 1992b] K. E. A. Silverman, E. Blaauw, J. Spitz, and J. F. Pitrelli, "A Prosodic Comparison of Spontaneous Speech and Read Speech", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 1298–1302, October 1992, Alberta, Canada.

[Spitz, 1991] J. Spitz, "Collection and Analysis of Data from Real Users: Implications for Speech Recognition/Understanding Systems", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 164–169, Morgan Kaufmann, February 1991.

[Svartvik and Quirk, 1979] J. Svartvik and R. Quirk, *A Corpus of English Conversation*, Liber, Stockholm, 1979.

[Svartvik, 1992] J. Svartvik, *The London-Lund Corpus of Spoken English: Users' Manual*, 1992, Distributed by the Norwegian Computing Centre for the Humanities.

[Swerts *et al.*, 1992] M. Swerts, R. Geluykens, and J. Terken, "Prosodic Correlates of Discourse Units in Spontaneous Speech", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 421–424, October 1992, Alberta, Canada.

[Taylor and Knowles, 1988] L. J. Taylor and G. Knowles, *Manual of Information to Accompany the SEC Corpus: The Machine-Readable Corpus of Spoken English*, January 1988.

[Umeda *et al.*, 1992] N. Umeda, K. Wallace, and J. Horna, "Usage of Words and Sentence Structures in Spontaneous Versus Text Material", in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 759–762, October 1992, Alberta, Canada.

[Winston, 1992] P. H. Winston, *Artificial Intelligence*, Addison Wesley, third edition, 1992, ISBN 0-201-53377-4.

[Woolbert, 1922] C. H. Woolbert, "Speaking and Writing – A Study of Differences", *Quarterly Journal of Speech Education*, 8:271–285, 1922.

Bibliography

[Ainsworth, 1988] W. A. Ainsworth, *Speech Recognition by Machine*, volume 12 of *IEE Computing Series*, Peter Peregrinus Ltd., 1988.

[Crystal, 1987] D. Crystal, *The Cambridge Encyclopedia of Language*, Cambridge University Press, 1987.

[Hardie, 1992] R. G. Hardie, *Collins Pocket English Grammar*, Harper Collins, 1992.

[Conway, 1935] R.S. Conway et al *On The Terminology of Grammar*, John Murry Publishing, 1935.

Appendix A

Repair Structures

This appendix lists the repair structures used by the system.

ART NOUN | ART ADJ NOUN

ADJ ADJ | VERB ADJ

PREP | ADV PREP

NOUN | ADJ NOUN

CONJ | ADJ NOUN CONJ

VERB PREP PREP | VERB

NOUN | VERB NOUN PREP ART NOUN

PREP ADJ NOUN | PREP ADJ NOUN

ART ADJ | ART

NOUN | NOUN NOUN

PRON | PRON

ART | ADV ART

ADJ NOUN | ADJ NOUN

PREP ADJ | PREP PREP ADJ

NOUN | ADJ PREP NOUN

PRON | ADV PRON

PRON | PRON VERB PRON

VERB | ADV VERB

VERB CONJ | CONJ PRON VERB

PREP | NOUN PREP

ADV | ADV ADV

PRON VERB VERB PREP VERB | PRON VERB VERB PREP VERB

CONJ VERB PREP ART | PRON VERB PREP VERB PREP ART

ADJ CONJ ADJ | PRON

PREP NOUN | PREP NOUN NOUN

PREP ART | PREP NOUN

NOUN | ART NOUN

ADV PRON | ADV PRON

VERB | PREP NOUN VERB

ADJ NOUN | ADJ NOUN

PREP NOUN PREP | VERB PREP NOUN PREP

VERB ADJ | VERB ADJ

ADJ NOUN | ART NOUN

CONJ | ART

NOUN VERB | NOUN VERB

ADJ NOUN | ADJ ADJ NOUN

PRON ADV | PRON ADV

ADJ NOUN PREP | ADJ

CONJ NOUN | ADJ VERB VERB ART NOUN

PRON VERB VERB VERB | PRON VERB VERB PREP NOUN

PREP VERB | ART

ART NOUN PREP ART NOUN | ART NOUN PREP ART NOUN

NOUN | VERB PREP NOUN

ADJ | ART ADJ

VERB ART NOUN | VERB ART NOUN

INTERJ | VERB

INTERJ | ART

INTERJ | NOUN

INTERJ | ADV

INTERJ | ADJ

INTERJ | PREP

INTERJ | CONJ

INTERJ | PRON

Appendix B

Seen Passage

This appendix gives the seen passage used in the seen passage analyses presented in this thesis.

now the first thing to tell you is the book the recommended book for this course < . > is that < . > software engineering the third edition < . > don't get the first or the third however cheap it is it's awe they're awful < . > it's this book here < . > it's eighteen or nineteen pounds but your all got plenty of money so you can all afford it < . > what i try to do on the course is that i don't exactly follow what's in that book < . > you should see this book as supplementary reading < . > i assume that you're reading the relevant sections and < . > occasionally i will point out the chapter you should read < . > that i don't have time to cover < . > i should also point out that you don't have to know everything that's in this book < . > you have to know the sections in the book that i cover in the lectures < . > and the sections that i point out each week < . > so there are bits in the book that i i i leave out altogether < . > and of course since your all very keen < . > erm here are some other books that you can go away and read as well < . > not don't buy any of these < . > couple of general software engineering books < . > in fact the the book by by collingham was the recommended book a couple of years back < . > the third one on the list the mythical man month common interesting contains interesting and funny stories about how software doesn't work < . > and why it falls over < . > and basically gives you some background information on why were here on this course < . > and the last book is software engineering economics < . > it's like a a typical economics books < . > it's a huge great thick book full of graphs and equations and and unusually for an economics book makes a lot of sense < . > but most of the stuff in this book is too far advanced for this course < . > but i'll refer back to that later on < . > right so there < . > some books you might like to look at < . > as well as books you should start now to think about what's happening currently in research < . > now you're moving into your second year and or at least a second year course

< . > and this particularly will be important when you reach your third year < . > or reach our third year courses < . > so in the library there are various err journals on software engineering and i've written three up up there < . > worth going along and seeing what the current thoughts are in research < . > this book thi this course is not abo about the research aspects < . > i'm just giving you some background < . > but again in particularly erm software engineering notes is quite an informal err journal < . > go along and read some quite funny stories about err again how software falls apart and all the strange little quirks that people design into their software < ..> the ques the first question to answer is what is software engineering < . > and here is a definition < . > software engineering is the technical and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates < . > so the first thing to notice about this definition is that it doesn't say that software engineering is about writing programs < . > in our courses that you took last year you were taught to program in modula two < . > if you like < . > writing programs is a very very small part of this term software engineering < . > so we're saying that software engineering is two part < . > technological < . > which is writing programs < . > interfacing to hardware < . > the whole bunch of sort of items like that < . > and it's also managerial < . > it's about managing people managing projects and managing your software < . > and hence that's why the reason that you are now doing this exercise for me in in managing your software < . > placing software where i tell you to place it < . > doing what i tell you to do < . > i'm trying to manage you in the production of a product < . > so it's important to to get this out of your mind < . > that all it is < . > is about writing programs < . > it's not about just about writing programs < . > and you'll get a chance to find out practical experience of some of these aspects when we come onto the group projects later on in the term < . > now just to put this into some sort of historical perspective < . > computer systems have advanced tremendously < . > i mean the the advancement is is quickening every day now < . > but the the advances have primarily been on the hardware front < . > hardware computer hardware has improved < . > size has increased < . > the speed has increased < . > and the reliability has increased < . > the hardware engineers have done a very good job < . > what we'll see in this lecture and maybe part of the next one is that really software hasn't advanced as swiftly as hardware < . > what this course is about is to try out and find out why that is so and what we can do about it < . > and i hope what you get at the end of the course is < . > that the understanding that we do need to be careful how we write our software systems < . > because more and more software systems are becoming life critical < . > erm we're writing systems that control nuclear power stations < . > items like that < . > and we can't just write programs and hope they work < . > because it gets a bit dangerous if our software falls over if it's controlling a nuclear power station < . > there is within the software industry a concept called the software crisis < . > and this has been due to various factors < . > people who are going to deliver software usually deliver it late < . > they usuall it usually costs more < . > it's usually very unreliable < . > it's usually very difficult to

maintain < . > by that i mean it's very difficult to change < . > when we write the program < . > and all the programs that you have written up to now were written and thrown away < . > in the computer industry people write programs and these programs continually change and evolve over the years < . > and this is a very important aspect of of software engineering called software maintenance < . > erm usually the last point when we write programs < . > it's not really what we user wanted < . > but we didn't know what he wanted anyway so we'll give him something and hope he's satisfied < . > now there are a whole collection of figures to support that these facts that the software projects do not satisfy normal delivery criteria and here is one set < . > this is a famous set of of figures of a study that was erm carried out in the united states < . > they had a software budget of six point two million dollars < . > and they were obviously going out to various suppliers of software < . > to try and meet the the needs on their requirements < . > and the figures in the pie chart are in to illic in illustrate the type of delivery that the software suppliers came up < . > and you can see that forty seven percent of the software that was delivered was never used < . > so they wasted three point two million dollars worth of money < . > twenty nine percent of their money went on software that was never ever delivered anyway < . > nineteen percent of their money went on software that was either abandoned or had to have more money spent on it < . > three percent of their money went on software that they could actually use after again had had major modifications < . > and the interesting figure is that two percent point one million dollars worth of software was that they could actually use as it was delivered < . > and this type of this is a fairly famous pie chart and is used throughout the computer industry to to show that maybe it's time we got our act together < . > maybe we should deliver software on time on budget and make sure it does what the user wants it to do < . > now in order to illustrate some of the aspects of why these things go wrong < . > before i get onto describing the if you like the central part of this course which is the the thing which is called the the software lifecycle < . > i've got a small series of drawings that try to illustrate why things tend to go wrong and < . > it's a a very simple tale about < . > if i take these off too quickly you should shout out cause i tend to not notice when you're still writing < . > so and this is again a famous set of drawings < . > now when we write programs or produce pieces of software for people < . > the first thing we have to do is we have to talk to them and ask them what they want < . > now that's fairly easy in your environment cause i tell you what to do < . > last year you had example sheets and you followed what the example sheet said < . > but if i was a software house i have to go and talk to the users to find out the users requirements < . > and most of the time the user doesn't know what he wants < . > so how on earth am i supposed to figure out what he wants < . > well that's the problem we're gonna try and tackle later on < . > once i've found out what he wants my software to do < . > it might be something to control the timetable in the university for example < . > i then set about designing the software < . > once i've desig got sort of a design < . > it's if you like the analogy between an archi some < . > if i'm gonna buil gonna build a bridge < . > the first thing i d do i don't start to buy the steel

and bolt it together < . > i shall have to design the bridge before i can build it < . > so i have to somehow design my software < . > and then once i've designed it i can actually finally get round to writing the program that implements what we want < . > so what i've got is a series of drawings that instead of software < . > it's for somebody who wants a swing < . > just an ordinary commonal garden swing < . > and the type of thing that happens is when you talk to the user < . > this is the type of thing that you might get a description from the user < . > they might describe that they want something hanging from a tree < . > and it's got three pieces of wood a rope < . > and i come away and i write up my one i think it requires < . > and if you like instead of writing out in english i've drawn a picture of what i think he wants < . > so as f < . > in my eye in my mind this is what i think the user wants < . > i then talk to my staff and say right this is what they want < . > can you come up with a a very simple specification of this < . > so my staff come up with a specification and this is what they come up with < . > you notice a subtle difference there err to what i've described < . > but never mind i don't notice that there's a subtle difference < . > so my staff have specified this is what the user wants < . > my staff haven't talked to the user < . > i've talked to the user < . > and then we pass it on to the designers < . > these are the people who are gonna if you like draw out the bridge or whatever we're gonna draw < . > draw out my swing in this case < . > so my designers after looking at the specification and my requirements design this < . > and again you can notice a subtle difference between the all the three pictures now < . > erm and remember my designers have only talked to my people who did the specification < . > who talked to me and i talked to the users < . > now it's no good the designers talking to the users because they don't talk the same language < . > so somewhere along the line communications at fault < . > and other things are at fault < . > then we give it to those strange breed of people who like to write programs < . > funny lot < . > and so they look at this design < . > and they write a program to do this < . > as all programmers are want to do they misunderstand < . > they think they know better < . > maybe it comes of being locked in a dark room for hours of end < . > so the programmers produce my software or in my case my swing to do this < . > but the delivery dates looming up and they realise that maybe this is not what the user wants < . > it's not got a lot of of functionality in there < . > so they sort of start quickly changing things hacking things about < . > and what they actually deliver to the user is this < . > at least you might get a little swing out of it < . > because you know we we were we spent a lot of time and and we were spending a lot of money < . > we had to get something out the door < . > because the user really wanted my swing now < . > the winter was coming on < . > and so we we take this to the users site < . > and of course err the user says hum that's not what i had in mind < . > what i actually had in mind was this < . > and we say well why didn't you tell me that in the first place < . > and of course he says i did and i say you didn't < . > so we argue < . > so what this is meant to show is that < . > if you're not careful < . > if you don't fully understand what you want in the first place < . > somebody tells me a half baked story < . > vague ideas in it < . > and then i try and formalise this and

pass it on to my err design specifiers and designers and programmers < . > you don't get that < . > you get that < . > and of course that's pretty hard to to convert from that to that < . > and the idea of this these whole set of drawings is that this is what's happening in the software industry < . > the industry wants something simple and the user wants something simple like that < . > and what we are delivering as software engineers < . > what companies are delivering are things like this < . > now you might think that that's a bit far fetched < . > but it's later on erm on the business and professional course < . > you'll have erm somebody coming along to talk about < . > erm different softw a topic called configuration management < . > and he will tell you stories real stories that show that you actually get things like this < . > the other problem in the industry because we we don't really know how to specify this system < . > is that it's also populated by very clever programmers < . > or people who think they're very clever anyway < . > and instead of delivering the simple tyre on a piece of string

Appendix C

Un-Seen Passage

This appendix gives the un-seen passage used in the un-seen passage analyses presented in this thesis.

okay < . > well < . > i'd certainly like to welcome you all this evening to durham < . > i know you've travelled quite a way from teesside and newcastle < . > it might be good to explain about the title of our talk this evening < . > and really i've split the talk into four parts < . > i'll give you a a bit of background information about erm our work within the school of engineering and computer science at durham < . > and i'll go on to erm give you some brief discussion of why i think processes are important < . > erm both in their own right and as basically a reaction < . > software processes anyway < . > as a reaction to the erm business process demands that are currently placed on i-t departments < . > erm i shall give you some of an introduction on the software process modelling research work we've done at durham < . > and erm go on to discuss how the approach can be more generally applied to process improvement < . > and erm well how it's changed management generally speaking on on business processes < . > within the school of engineering and computer science < . > in the computer science group there are two research groups < . > i work in the centre for software maintenance < . > which was established in nineteen eighty seven < . > erm over that period since since eighty seven we've had some key achievements < . > basically < . > we've established here at durham an international workshop on an annual basis < . > which now gets one hundred and sixty people attending it < . > not just researchers but practitioners interested in in software maintenance < . > this is a really hot topic in software engineering < . > and we we have also established a journal of software maintenance < . > both research and practice < . > and erm we- we've been very active in the research field < . > erm we've finished recently an esprit two project on reverse engineering < . > focussing really on re-documentation < . > called redo < . > and we've been very fortunate indeed to have three d-t-i research grants on the information engineering advanced technology programme < . >

erm one of which i'm going to describe erm tonight < . > these projects are very important to a university research department < . > because they allow us to collaborate extensively with industrial partners < . > so we move ourselves from an ivory tower pure research approach and actually try and apply research in an industrial context < . > which is quite a challenge in it's own right < . > we've also got a number of research students < . > case award type studies < . > and again < . > the focus in the centre for software maintenance is really not to look at software maintenance on small student type programs of one hundred or maybe one thousand lines of code < . > but major erm industrial type problems that have have mill- millions of lines of code < . > we're looking at sort of working with the bank inspector < . > and with m-o-d type software < . > i personally look at command and control systems for for the naval fleet < . > these are are old systems < . > twenty years plus < . > so they've been extensively maintained over this period of time < . > here is a copy of a description of our research strategy there < . > and basically < . > what we've recognised is that software maintenance must support the needs of the business and be consistent with the overall organisational policy and strategy < . > this has been one of the the basic drivers for the the research i've done on process modelling < . > recognising that software maintenance isn't just a a technical activity < . > but it- it's really an an organisational and management process < . > trying to represent that in in a graphical model is quite a challenge < . > and the other thing that you're probably not or maybe you are aware of is that software maintenance is quite a a key topic for companies these days < . > because i-t consumes so much of the available i-t resource in any company < . > and some of the old surveys indicated that twenty to seventy percent of the available i-t resource was working on software maintenance in one form or another < . > and and some companies that i've had dealings with recently have near ninety five percent of their available software engineers working on software maintenance < . > you know this is quite a a strategic problem < . > and and erm here we feel that research should really focus on development methods < . > looking at rigourous development methods < . > and we do have an interest in development < . > but we look at it from the angle of maintainability < . > i-e trying to tell erm development teams really how they can improve the maintainability of software < . > for the the whole of the the product lifecycle < . > rather than just as a development project < . > if you think about a normal project of perhaps say two years development < . > that project could be inexistent for over twenty years < . > so maintainability is a key issue < . > and this is something that we're addressing with some of our our newer research projects < . > more specifically my my research is has the overall aim of developing and applying process models < . > which facilitate the management of software maintenance < . > and as malcolm mentioned it's due to erm to be completed this summer < . > and so we're we're quite a long way in- into our experimental trials of of our of our process improvements and our our process models < . > and we've been trying to develop these process models over the course of the project < . > and the key factor i think we must remember is that these aren't process models being developed just in a a computer science laboratory < . > these are process models

being developed of erm current practices within an industrial context < . > and we are looking at process improvement in an industrial context < . > so we're needing < . > you know < . > two to three years to to basically measure the efficiency of our approach < . > and that's the sort of timescale that you're really considering if if you want to measure process improvement on a large scale < . > to name a specific objective of my research < . > which is on that foil < . > erm we have different partner responsibilities < . > and at durham i've i've been concerned with developing the best practice model for the software maintenance process < . > and erm basically working in conjunction with our other partners < . > ferranti international and lloyds register of shipping < . > attempting to meet ferrantis management goals in on a real maintenance project < . > and lloyds register with their background in assessment and survey < . > have actually been erm assessing through a a measurement scheme the efficiency of of our approach < . > so this isn't sort of process improvement in a-t-q-m sense < . > this is basically actually being measured in effectively a software engineering trial < . > and if you consider that the project that i'm i'm looking at has been going for over twenty years < . > then the evolution < . > within the company < . > of the management approach is quite extensive < . > and so to come up with any process improvements < . > and and have it measured is is quite a challenge < . > when you are not necessarily experienced in in that particular industrial domain < . > some work that lloyds sorry ferranti have been doing < . > has been < . > basically < . > having got a defined process out of this project < . > looking at further opportunities for method development < . > and also tool support for their existing software maintenance < . > this is what we regard as more of a structured approach < . > erm rather than the sort of ad-hoc use of methods and tools < . > that erm whatever happens to be perhaps fashionable at the time < . > so they're using their their process and looking at weaknesses in their existing process < . > to drive their their strategy towards erm perhaps case tools < . > i've mentioned that erm maintenance a lot < . > and perhaps thought well perhaps we have to spend that much amount of effort on maintenance < . > well i suppose that basically the i triple e definition of maintenance < . > and as you can see < . > erm when i talk about maintenance < . > erm i'm talking not just about corrective bug fixing or patching of software < . > but also perfective maintenance < . > erm enhancing software to meet new functional requirements < . > we also include adaptive maintenance < . > which is basically porting software to new hardware environments or adapting software because the operating systems etc have changed < . > and a fourth category of maintenance which isn't really performed that much < . > is is preventative work < . > which is done in order to prevent malfunctions < . > or or basically to improve maintainability < . > erm and so the only thing that i i really do exclude from software maintenance is is basically brand new development from basically a clean sheet of paper < . > so that's a bit of the background < . > if i i move on to erm some process viewpoints < . > i thought it would be useful basically to erm discuss why sort of interest in the process has arisen < . > and really looking back a few years i think its because of the emphasis on on effectively the quality challenge < . > you know < . > in terms

of trying to improve software quality < . > and i guess two aspects there are < . > meeting or satisfying customer requirements < . > in time < . > on budget < . > also basically erm getting the right functional correctness there on day one < . > and that's not a really bad starting point to consider software processes < . > in the nineties now we seem to have a a economic argument < . > that says that if you adopt a a traditional approach < . > which relies on intensive testing in detecting and eliminating erm defects prior to to sort of delivering software < . > whether it be a new development or maintenance project < . > then you're not really restricting or or preventing that that erm problem arising again < . > and the only way in which you can actually improve the process is is is to detect and eliminate the source of the defect < . > and this really call- calls for a repeatable process < . > which i'll come back to later on in this talk < . > but just latch onto the words repeatable process < . > this is a significant style < . > but basically that allows you to erm drive an improvement programme < . > because it allows you to have realistic monitors and management metrics < . > for process and product metrics < . > to actually improve software quality < . > so you can actually get a handle on this very sort of vague attribute < . > okay < . > so i haven't got the ideal definition of a process < . > but rather erm a more colourful erm dictionary definition of a process < . > erm i think the key notions that that you've got to recognise < . > if you're going into the process modelling business < . > is the idea of connection < . > the idea of of carrying out actions < . > and and the idea of progress < . > now that has caused quite a lot of problems in in sort of static process modelling < . > in in sort of erm recognising what progress is < . > and erm really the way we've approached that is to follow up some of the themes which haven't been driven by the i-t community < . > but more through the business community < . > directorate < . > and things like this < . > where there has been concern with with such things as business transformation < . > and business process redesign < . > and we've been really taking a lead to some extent in sort of both asking the i-t community for for support in analysing what the sort of what the business processes are < . > and erm that is erm something that is basically causing a realignment of the the management roles < . > certainly software managers roles < . > between < . > you know < . > what we effectively regarded as a black box technical activity < . > such as software maintenance < . > and why it's of strategic importance to the business < . > some of the things that have come out of the the work in the wider community are shown there < . > erm basically what much of the the work has recognised is that erm you need actors < . > and these are either individuals groups units < . > generally these actors as such have objectives < . > responsibilities < . > competence levels < . > experience function structure < . > these sort of attributes which are are not normally shown in in sort of your typical analytical model < . > that you perhaps do as part of your systems analysis work < . > in perhaps analysing the requirements for a new system < . > and so this is has really stretched our analytical techniques < . > so that say perhaps business analysts can actually represent some of these characteristics which we need to describe < . > if we're we're describing a a process < . > whether it be a business process or a

software process < . > and the other factor is that not all these factors suit humans < . > we have erm tool support and and other types of agents which we have to recognise < . > if if we're perhaps doing a model of current processes or current practices in an organisation < . > and in terms of what i mentioned about progress < . > on the other slide < . > erm basically what the businesses recognise < . > is that business processes have a purpose < . > and therefore the software processes should have a purpose < . > and erm this is really an important aspect of our work < . > which we have to pick up on an- another < . > well using another technique < . > okay < . > so that's quite a lot of words < . > but graphically it looks something like that < . > and this sort of business perspective has been very useful < . > and provided a lot of leverage in our our process modelling work < . > and the key things there are some of the models that are available from day one < . > like the organisational hierarchy < . > which you can sort of get from an organisation chart < . > you can look at peoples procedures < . > there are often departmental operating procedures < . > things like that < . > one of the the real challenges is really to put that to one side and try and work out how the individuals act together < . > you know < . > the team work that and the networking that goes on at a departmental level < . > or the cross function within an organisation < . > these are within the software process < . > and the the key that is to understand what roles people play < . > and what actions they perform when they play those roles < . > and again from from what i mentioned abo- about a purpose < . > you've got to look at what the common goals are < . > and erm these goals do vary depending upon what level of the organisation you talk to < . > so when you start to sort of do your information gathering < . > you've got to basically erm to do both a depth and breadth study of the of the the management objectives < . > if you just concentrate on something like software maintenance < . > such as a software maintenance manager < . > you end up with basically a technical black box < . > which he doesn't really understand < . > which i guess sort of fits in in that bit there < . > now just for completeness sake i've put some of the other organisational infrastructure around it < . > to make it more of a a successful model < . > i feel < . > but in practice < . > very little of that infrastructure exists < . > a lot of software maintenance is carried out without a plan < . > a concept of control within software maintenance such < . > as request control < . > change control < . > and release control < . > are are sort of well there to in varying degrees < . > and then certainly erm there is not much use of even recognised software engineering models < . > like reliability models < . > cost models < . > used in the software maintenance arena < . > so planning and control erm erm from a management perspective are generally missing < . > if if you look at at a typical software maintenance model < . > to sort of put all that lot together < . > you can see that we are talking about a very different type of model < . > than a typical activity model or organisational model or data model < . > and we have a few sort of tools up our sleeve < . > erm one is a sort of static process description < . > and that's perhaps the most popular because it's the easiest to do < . > you just need an pencil and paper basically < . > erm what you do there is you're doing a functional

model < . > basically what you try and ascertain are are the process elements < . > the actual work that's being performed and the information flows between those process elements < . > so you end up with like a a data flow diagram type approach < . > which is quite popular < . > and also it's a good useful tool for actually trying to erm talk to people < . > to to actually verify that you really represented or described their process < . > the structural representation is slightly different < . > in that you look at erm the the necessary agents < . > the organisational entities needed to support those process elements < . > and the communication path ways < . > whether they be logical or physical < . > between those organisational agents < . > erm i'll pick one more < . > the the behavioural aspect < . > that is looking at erm when process elements are performed and how they're performed < . > you know < . > whether you've got feed back loops < . > iterations < . > decision conditions < . > and entry and exit criteria < . > so things like an e-t-v-x type model < . > entry path validation exit criteria model < . > is a way of of basically showing the behavioural model < . > and some of those can be used as simulation models < . > we have done just a little bit of work on that < . > erm there are other model frameworks erm which have been used to to describe processes and measurement < . > we we have tended to look at things like the the goal question metric paradigm from erm john collingham and dave reagan over in the states < . > which erm provides basically a framework < . > so you design your metrics < . > it allows you to focus first on on the goals to be achieved < . > so this fits in well with that business perspective < . > focus on the goals < . > and then you define the questions which you need to answer to get some insight into those goals < . > and finally erm you devise the metrics which which could be used to answer those questions < . > and really to do that < . > the model becomes the process model becomes very useful < . > because you can look at process elements and the products < . > or you can look for new or novel process process prod- product metrics < . > assessment and improvement frameworks < . > i'll i'll show you one that you'll find useful < . > well i find useful < . > this is the erm software engineering institute erm c-m-m model < . > capability maturity model < . > which is developed at durham university < . > but it goes back a long long way < . > like to the nineteen twenties in quality technology < . > in terms of lots of five level models < . > that are used in in manufacturing engineering and areas like this < . > so it's not really new < . > it's just perhaps new in in quality in in software engineering terms < . > now the general thing about this model is that people look at it and say < . > well we have managed software processes < . > we must be up at level four < . > you know < . > okay < . > we erm we measure our process < . > we have some quantitative evidence about how much resource is used < . > this sort of thing < . > and yeah < . > okay < . > technology changes < . > this is a problem < . > you know < . > and we we're trying to improve our software < . > you know < . > prevent problems occurring < . > well everybody sort of intuitively thinks they are up here < . > but in actual fact most people are down here < . > i'll show you that on the next slide < . > but basically what what this framework does is is provide you a framework for identifying five levels < . > and then narrowing the scope for the

improvement activity < . > and that's erm of significant benefit over a total quality management approach < . > where you you generally can go off in any direction < . > providing you can prove the process < . > there is some doubt whether it all comes together at the end < . > to actually achieve both your business goals and indeed improve the process < . > because there is not much in the way of a measur- measurement framework there < . > this does allow you the ability to put successive layers down < . > which gives you the foundations for continuous process improvement < . > very much along the japanese lines again < . > but erm just to show you where everybody is < . > erm i took this slide from some proceedings from i-p-s-s european international conference < . > well < . > what basically that shows in the top graph < . > is that one hundred and thirteen u-s space sites < . > and the f-b-i you've got to realise effectively has a department of defence type contract < . > for the united states government < . > so these are your companies like field aircraft corporation < . > companies of that sort of order < . > and that shows the the sort of distribution < . > all this lot here are in level one < . > with the odd few in in level two < . > and maybe some up in in the top end of level three < . > and the japanese are not so good either < . > they've had a a look at erm basically two hundred japanese type software producing units < . > and again < . > look at the distribution in level one < . > erm european sites < . > not many european sites have been assessed < . > but those that have have generally followed the the bootstrap questionnaire approach < . > which is a slightly more sophisticated and refined version of the f-b-i model questionnaire < . > because what they recognised in in in europe is that not all companies are departments of defence contractors < . > you know < . > we have different market domains < . > and there are different size software producing units < . > and so it tried to produce a more general organisation model < . > it's quite encouraging as far as i'm concerned < . > to see that erm we've got so many level two organisations