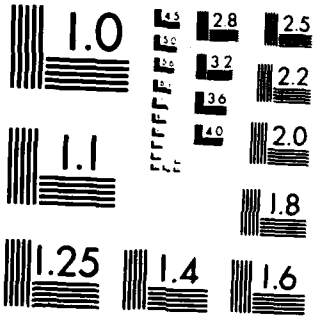
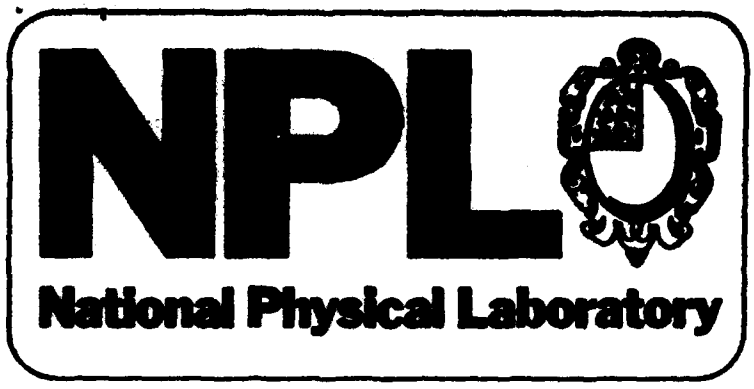
AD-A136 998 A SPEECH CONTROLLED INFORMATION-RETRIEVAL SYSTEM(U) 1/1
NATIONAL PHYSICAL LAB TEDDINGTON (ENGLAND) DIV OF
INFORMATION TECHNOLOGY AND COMPUTING M P COOKE JAN 83
UNCLASSIFIED NPL/DITC-15/83 F/G 5/2 NL





NPL Report DITC 15/83
January 1983

A SPEECH CONTROLLED INFORMATION-RETRIEVAL SYSTEM

by

M P Cooke

NPL Report DITC 15/83
January 1983

N A T I O N A L P H Y S I C A L L A B O R A T O R Y

A Speech controlled information-retrieval system

by

M P Cooke

Division of Information Technology and Computing

Abstract

This report describes a speech controlled information-retrieval system. The problem of speaker-dependence is considered. The system exhibits a simple, intelligent man-machine interface.

A

Contents

1. Introduction

WISPA (Word Identification in Speech by Phonetic Analysis) is a software research tool developed at NPL to evaluate various strategies for the recognition of continuously spoken speech (See reference [1]). The acoustic signal is preprocessed by a piece of hardware known as a Speech Input Device (SID Mk. 3) (See reference [2]). Although intended primarily for research, WISPA provides an interface through which an application program can send commands and receive messages. One particular application viz. An Information-Retrieval System Controlled by Speech, is described in this report.

The primary aim in creating this application was to provide a demonstration of speech recognition for two open days in the Division of Information Technology and Computing at NPL. The information held by the system could then describe the research carried out by the division, and provided visitors with an opportunity to use speech themselves to access useful information.

The demonstration afforded an opportunity to investigate the problem of recognising speech from a large population without the possibility of training the system for each speaker. This is generally known as the multispeaker problem.

The system was demonstrated to many different speakers with varying degrees of computer experience. It therefore had to be simple to use, with a free flowing interaction befitting the use of speech as a means of communication. Attention was given to producing a sensible, intelligent man-machine interface, described in section 5 of this report.

2. Recognition Strategies

WISPA provides three speech recognition strategies enabling recognition of

words spoken in isolation,

words embedded in phrases,

and continuously spoken speech

Although the ultimate goal is to recognise continuous speech from many speakers, it was decided to remove the extra dimension of continuously spoken speech for the demonstration in order to study just the multispeaker problem. The information was therefore accessed by sequences of single words or short phrases.

3. System structure

Figure 1 shows the structure of the information-retrieval system. It is essentially tree-structured, with each node representing a decision point for the user. To keep the interaction simple, the tree has just three levels, so that two choices are made to reach the desired information. In a larger application this could easily be extended to provide more levels whilst keeping the same overall structure. In fact, this representation is useful in implementing efficient syntactic checks as described in the section on syntax.

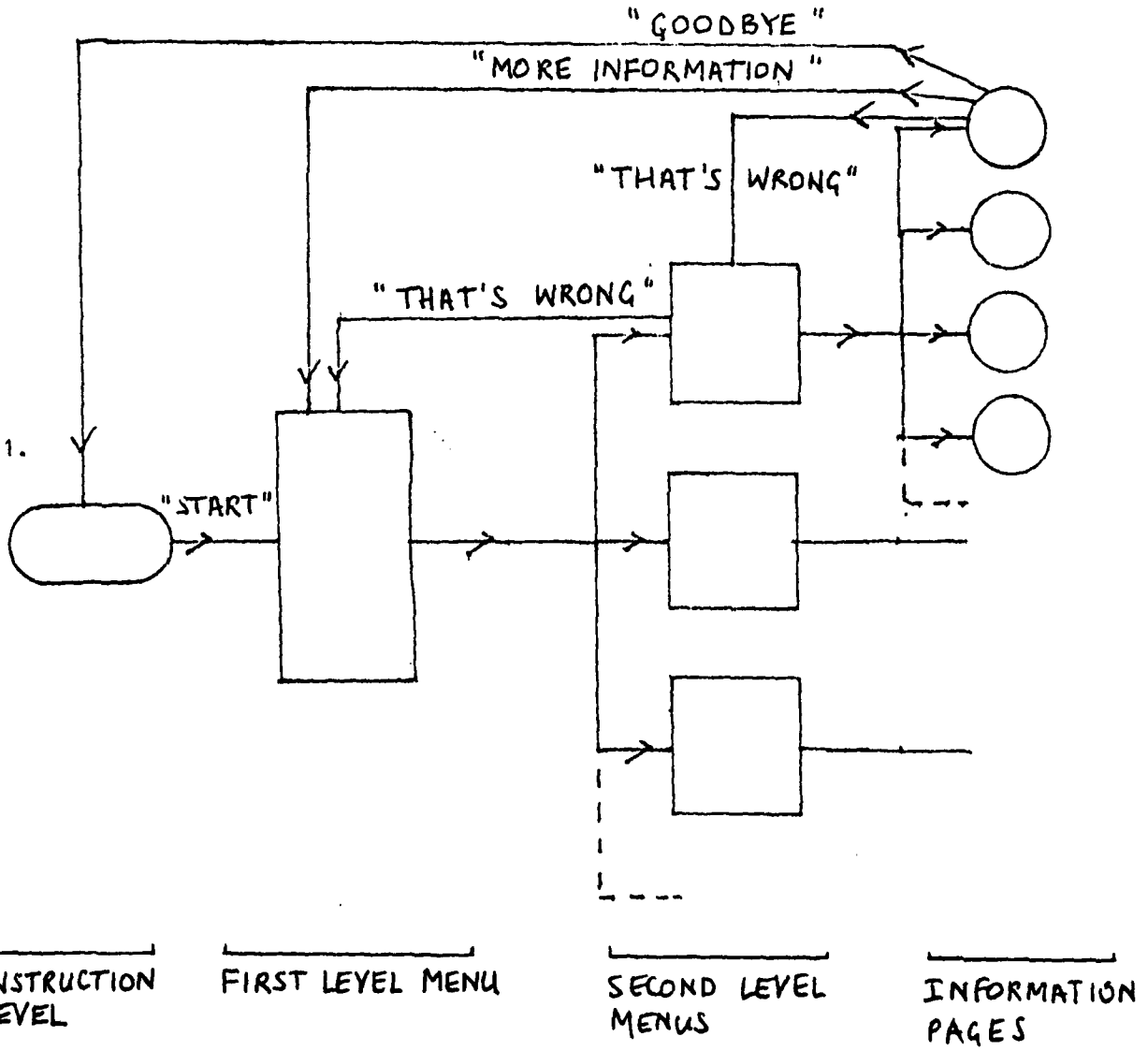


Fig 1.

3.1 Program design

The program consists of a parser and a display facility. The action of the applications program is to instruct the speech recognition hardware to listen for utterances and to receive decisions based on the WISPA software's perception of such utterances when they occur. These decisions cause the parser to change its state. With reference to figure 1, these state transitions correspond to well-defined movements between nodes of the connected graph. The parser's state and the particular word recognised together define a page reference which causes the page relevant to that point in the interaction to be displayed on the screen. The program design ensures that the page display routines are kept separate from the parser, to the extent that the main part of the program consists of the parsing mechanism.

3.2 Syntax

3.2.1 Finite State Machine Representation

One method of implementing the parser is by using a finite state machine. Each state of the machine corresponds to a node in the directed graph of figure 1. This method is used by Levinson in research to find the effect of *syntactic analysis on word recognition* accuracy [3].

WISPA currently provides facilities to check the syntax of phrases which belong to a regular grammar, where a grammar is said to be regular if the phrases defined by the grammar can be accepted by a finite state automaton.

However, this has several disadvantages. Firstly, for a finite state machine with n states, one would require an n x n matrix to hold all possible transitions from one state to another. This information could be held more compactly in a sparse matrix representation at the expense of access time and ease of programming. Also, because the syntax defines which subset of the total vocabulary is allowed at a particular stage in the interaction, key phrases such as "that's wrong" which are allowable at many points, have to be present in each subset.

3.2.2 Pushdown Automata

These difficulties are caused by modelling the system as a finite state machine. A more powerful formalism often used in syntactic analysis is the pushdown automaton. These machines accept a class of grammars known as context free grammars and correspond to Type-2 grammars in the Chomsky classification. Without explaining the language theory behind these formalisms (which can be found in reference [4]), it is sufficient in this application to note that the main difference between pushdown automata and finite state machines is that the former possess an explicit memory in the form of a stack, whereas the latter use their states to remember previous inputs. The finite state machine is a subset of the pushdown automata. The use of a stack enables the recognition of structures such as arithmetic expressions with arbitrary levels of nesting. Such structures cannot be recognised by any finite state machine.

To see how a pushdown automaton is useful in this application, consider the information-retrieval system to have a hierarchical structure as in figure 2.

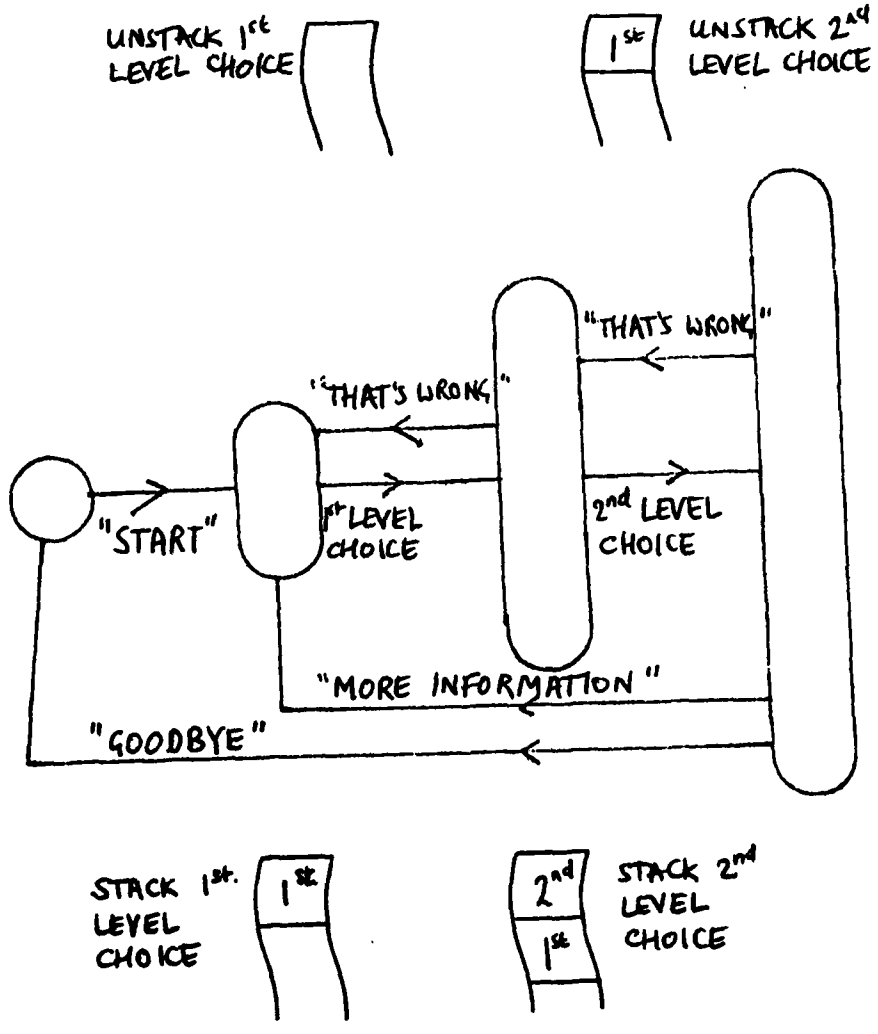


Fig. 2

In progressing from level 1 to level 2, the actual choice made is pushed on to the stack, and similarly between level 2 and level 3. If a "that's wrong" is recognised, the top item is popped off the stack and the new stack top conveys the information about which sets of words are allowed next.

A total of 4 states are needed to model this process, which compares favourably with the 16 states needed in the finite state machine representation.

3.2.3 Automating the process of syntax generation

Software tools exist for producing a finite state machine representation of a system from simple rules ([5]). There is no reason why this stack-based approach cannot be similarly automated. The application of syntactic constraints is an important part of any speech recognition system, and with larger vocabularies becoming necessary, it requires an efficient means of implementation.

4. The multispeaker problem

A degree of speaker independence can be introduced into a speech recognition system during the training phase or at recognition time. The NPL approach of recognising speech from phonetic-like features extracted from the acoustic signal means that training and recognition is attempted at a level where some of the speaker variability has been removed from the signal.

4.1 Training

WISPA is trained by combining several utterances of the same word into a single template. The method used for training is important because if the system has a 'bad' training template to begin with, it is unlikely to be very successful during subsequent recognition trials. This leads to the question of what constitutes a 'good' template. Obviously, the definition of good is related to the application. For a single speaker system, a 'good' template is one which encompasses all the variations which the speaker will ever use producing the utterance. In this application however, a 'good' template is one which contains within it all the differences a whole range of speakers introduce when saying the word, but without permitting enough variety to make all templates look too similar.

In the absence of extensive research on optimal merging strategies for WISPA, each of the training templates for this application consist of 2 utterances each from four speakers. Tests showed that the performance of the system for the original trainers did not deteriorate when other speakers utterances were merged into the initial templates.

4.2 Runtime adaption

A system which can adapt its recognition procedure to favour the voice of its current user will be less speaker dependent than one which performs its recognition statically. There are important factors to be picked up whilst the system is being used which could serve as cues to recognition, or limit search time. For instance, monitoring the speed of articulation continuously could lead to a faster time warping algorithm by restricting the amount of overlapping of words.

A run time adaptive system relies heavily on accurate feedback from the user. For instance, a system can only adapt in some circumstances if it knows whether or not it is recognising correctly. The user must then supply the system with such messages as "that's wrong", with the absence of such a phrase to mean tacit acceptance. The system can then build up a class of misrecognised utterances, and, by introspection, analyse why it is recognising these particular phrases wrongly.

In this application, a low level of run time adaptation is built into the interaction. This is described in the section on error correcting (5.2).

5. The man-machine interface

The interaction in a good man-machine interface should be designed to allow anyone to use the system, in a natural manner and without the need for assistance. Speech is, of course, ideal for this situation, eliminating the need for a keyboard. However, the familiarity of speech to man also presents problems, because the user will attempt to use all the short-cuts he has learnt to which other humans are tolerant. The interaction used in the demonstration is described below, each paragraph explaining a particular feature.

5.1 Simplicity

On arriving at the demonstration, the visitor is presented with three simple instructions enabling him to get the system started. Having said "start" the visitor is given a brief description of the information in the system, & told how to correct misrecognitions by using the phrase "that's wrong". Following this, the interaction is menu-driven.

At any stage in the process, the user should know where he is, how he g there and what to do next. This is accomplished by formatting each leve differently, with some pertinent reference to the previous utterance.

eg 1 Having made an initial choice of "information processing"

the system responds with

OK, you've expressed an interest in information processing.

eg 2 On the final level when the information page is displayed, a page reference consisting of the first and second level choices which causes the selection of the page is printed.

eg 3 If a "that's wrong" is heard, the system will give a different response than would normally be got by moving back one level.

Utterance - INFORMATION PROCESSING

System response - COMPUTING STANDARDS

Utterance - THAT'S WRONG

System response - Sorry, repeat your choice please.

These features make it easier for the user to follow the interaction. See section 5.5 for a typical interaction.

5.2 Tolerance to 'meaningless' noises

Typically, a demonstration environment is noisy with the user competing with many other speakers, doors banging, coughs and splutters. If the microphone picks up some signal due to one of these extraneous sounds, the sensible course of action is to ignore the resulting decision. This provides a reject option which causes no change in the interaction with the exception of instance 3 below.

This application has three rejection criteria.

1 - If the number of features detected in the spoken command is less than half the the number of features in any of the trained templates under consideration at the time, then reject. This eliminates short bursts of noise such as a cough, mutter, chair scaping, door slamming and a large range of other noises

2 - If the number of features detected in the spoken command is more than twice the number of features in any template currently considered for recognition, then reject. This has the effect of ignoring continuous chatter produced, for example, by someone getting too familiar with the system and attempting to hold a conversation with it.

3 - If the lowest penalty accrued in matching an utterance with a particular set of templates is more than a tolerance multiplied by the length of the template, then reject. Thus, if someone mistakenly says a word which is not allowed at a particular stage, it should produce a bad enough fit to be rejected. After such a reject, however, the tolerance can be relaxed for the next utterance to allow words through which were allowed but which produced such a bad fit as to be rejected. These should get through next time. This is a low level adaptive process as the tolerances are controlled by the particular user's quality of utterance.

A rejection is signalled to the user as "PARDON !!" followed by a short delay.

5.3 Error correction

A subset is a collection of words from the vocabulary. WISPA is limited to providing statically allocated subsets i.e. the subsets have to be defined during training. In this application, when the user says "that's wrong", the sensible error correcting protocol would be to delete the misrecognised word from the current subset. This implies altering the syntax of the system during recognition.

In this application, at the final level of the interaction, the misrecognised word is deleted from the current subset. The user is presented with a menu which reflects this deletion.

5.4 Speak - Don't speak

To create a robust system, it is advantageous to limit the occasions on which the user can speak to the system. In other words, the speech recognition equipment is activated only when necessary. This is a constraint on the user during the interaction, but

(i) it ensures that the user has time to read all the information,

(ii) it causes the user to read the instructions,

(iii) it prevents the system from picking noises up at times when the user is unlikely to want to speak to the system.

Initially, this slows down the interaction with the aim of ensuring that the user has time to read all the instructions associated with each level. On a second run through the system, it is possible to remove these delays, allowing a fast interaction.

The user is informed that the system is listening by the message

"You may now speak"

appearing on the screen.

5.5 Typical interaction

Comments in brackets are not displayed.

USER : "START"

SYSTEM : The information in this system describes the work of the Division of Information Technology and Computing at the NPL.

Please hold the microphone close to your mouth.

If the system misrecognises your choice, then say "that's wrong".

In a moment you will be able to speak.....

{ new page }

Which of the following topics interests you ?

INFORMATION PROCESSING

COMPUTING STANDARDS

MATHEMATICAL SOFTWARE

{ delay first time through }

You may now speak

USER : "INFORMATION PROCESSING"

SYSTEM : OK, you've expressed an interest in

INFORMATION PROCESSING

Within this field of research, the division investigates the following topics. which one would you like to receive information about ?

SPEECH RECOGNITION

COMPUTER VISION

MULTIPROCESSOR RINGS

EDUTEXT

{ delay first time through }

You may now speak

USER : "SPEECH RECOGNITION"

SYSTEM : Page Reference :- INFORMATION PROCESSING - SPEECH RECOGNITION

NPL research aims are to develop systems that will accept speech spoken in a natural way without the need for pre-training.

The system concept is to fully utilise the natural language disciplines such as syntax, context, semantics etc.

The conditional feature analysis protocols that are being developed are more akin to understanding rather than recognition.

Project leader : Brian Pay Exhibits 22, 28

{ after a delay long enough to read the page }

If you would like information on another topic, say "MORE INFORMATION"
If you have all the required information say "GOODBYE"

USER : "MORE INFORMATION"

An example of error correcting protocols is shown below.

SYSTEM : Which of the following..... etc. (without delays)

USER : "COMPUTING STANDARDS"

SYSTEM : OK, you've expressed an interest in

MATHEMATICAL SOFTWARE

etc.....

USER : "THAT'S WRONG"

SYSTEM : Sorry, repeat your choice please .

INFORMATION PROCESSING

etc....

6. Statistics gathered

The following statistics were collected :-

Number of speakers	=	110
Total number of commands	=	849
Total number of misrecognitions	=	112
Total number of rejects	=	276

As the most important requirement in gathering the statistics was to not interfere with the interaction, they were collected automatically instead of by hand.

The number of misrecognitions mentioned above corresponds to the number of times the phrase "that's wrong" was recognised. Thus it serves as a crude approximation to the actual number of misrecognised commands as it relies on the user indicating the misrecognition. However inspection of several interactions gave the impression that users were using this command whenever a misrecognition occurred.

The number of rejects measured takes no account of the cause of the rejection.

For each user, the following statistics were kept :-

Number of commands

Number misrecognised

Number of rejections

Total penalty accrued

The results of the individual users statistics are given in figures 3 - 5 .

Distributions

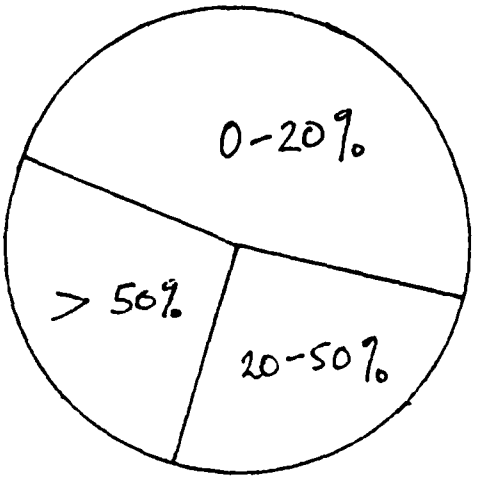


Fig. 3 % rejects per user.

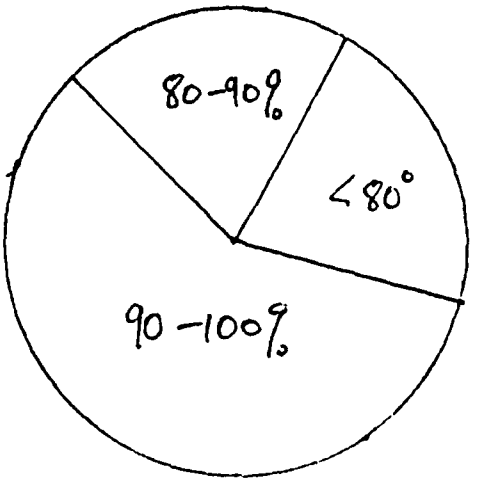


Fig. 4 % recognition rate per user.

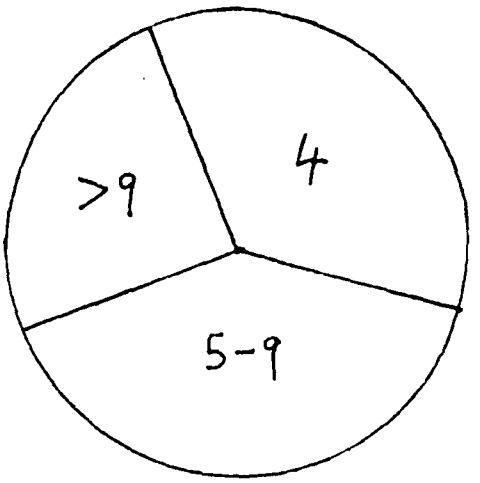


Fig. 5 Number of commands/user.

The following is a synopsis of the statistics :-

The average number of commands spoken by each speaker = 8

Of all the sounds heard by the system, i.e. user commands, background speech from other people and environmental noises, 25 % were rejected.

Of the sounds not rejected in this way, 87 % were recognised correctly.

The last figure should be compared with an average 95 % correct recognitions when the system was used by speakers who trained the system.

6.1 Background noise

It was observed during the interaction that background noise accounted for the majority of "pardons". This is a good measure of the robustness of the system, in rejecting extraneous signals in a sensible manner.

References
---

[1] Yardley J P (NPL) (1981)
"WISPA : A System for Word Identification in Speech by Phonetic Analysis"
Ph. D. Thesis, Dept. of Electrical Engineering, University of Essex, May 1981.

[2] Rengger R E & Manning D R
"Improvements in or relating to Speech Recognition systems"
British Patent Application No. 8136679 , 4 Dec. 1981.

[3] Levinson S E (1977)
"The Effects of Syntactic Analysis on Word Recognition Accuracy"
The Bell System Technical Journal, Vol. 57, May-June 1978.

[4] Hopcroft J E & Ullman J D (1979)
"Introduction to Automata Theory, Languages and Computation"
Addison-Wesley.

[5] Lesk M E (1975)
"LEX - a lexical analyser generator"
CSTR 39, Bell Laboratories.